

Guía de Funcionamiento





Guía de Funcionamiento

Indice

1. El ordenador y su instalación

El objetivo de este manual es ayudar al usuario, ya sea principiante en el mundo de los ordenadores o no, a descubrir las posibilidades de su ordenador INVES SPECTRUM+.

Al desembalar su INVES SPECTRUM+, deberá encontrar:

1. Este manual.

2. El ordenador, que dispone en su parte posterior de tres conexiones hembra marcadas como 9V CC IN, EAR y MIC, un conector de TV., un conector macho de 9 pines para joystick y un bus de expansión destinado a la conexión de accesorios opcionales.

3. Una fuente de alimentación, que sirve para convertir la electricidad suministrada por la red, en la que utiliza su INVES SPECTRUM+. Si desea emplear su propia fuente de alimentación, habrá de proporionar 9 voltios de corriente continua con un amperaje de 1,6A sin regulación.

4. Un cable de antena de 2 metros de longitud aproximadamente, que se utiliza para conectar el ordenador al televisor.

5. Un cable de unos 75 cm, de longitud con doble clavija de 3,5 mm en cada extremo, para la conexión del ordenador al magnetofón a cassette.

6. Una cinta que contiene programas de ejemplo.

Necesitará también un televisor. Su ordenador INVES SPECTRUM+ puede funcionar sin dicho aparato, pero así no podrá ver lo que está haciendo, lo cual sería una limitación bastante importante. La sintonización se hace en el canal de UHF. Si dispone de un televisor en color podrá disfrutar de la señal de color que proporciona su ordenador INVES SPECTRUM+. Si trabaja con un televisor en blanco y negro, los distintos colores que puede generar su ordenador aparecerán como seis tonalidades diferentes de gris. No obstante, el ordenador funcionará tan bien como con un televisor en color.

Los componentes del sistema deben interconectarse tal y como se indica a continuación:

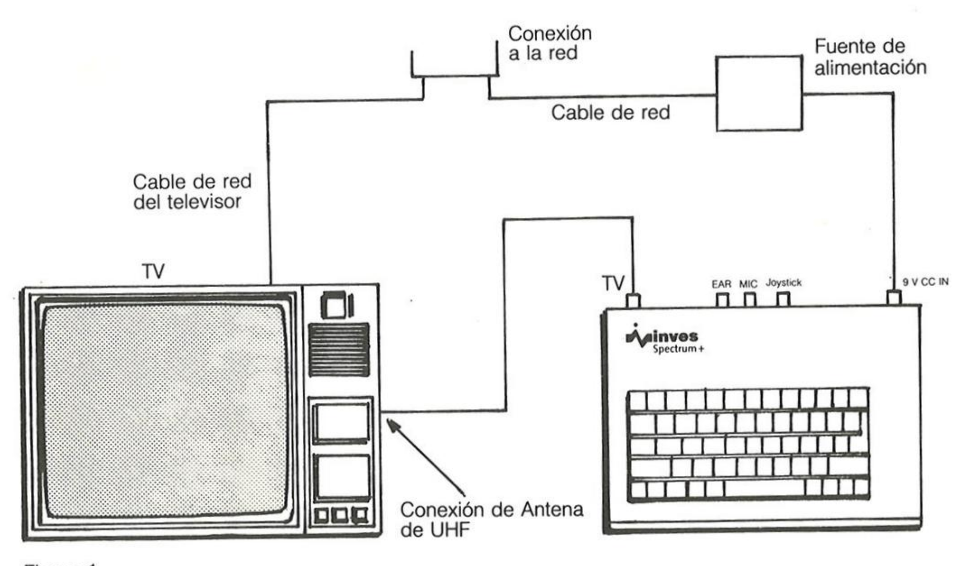


Figura 1

Si su receptor de televisión dispone de dos conexiones de antena, marcadas como UHF y VHF, deberá utilizar la de UHF.

Conecte la fuente de alimentación y encienda el televisor. Sintonícelo en uno de los canales que no utilice normalmente, hasta que aparezca en la parte inferior de la pantalla el siguiente mensaje:

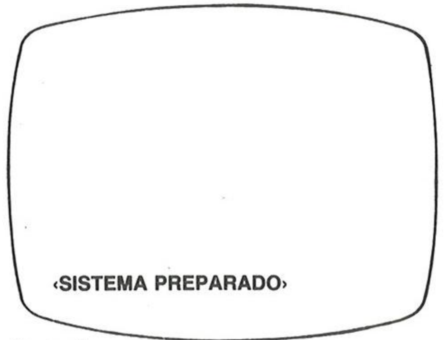


Figura 2

Cuando utilice el ordenador deberá ajustar el mando de volumen de su televisor al mínimo.

Siempre que desconecte su INVES SPECTRUM+ perderá toda la información almacenada en el mismo. Una forma de conservarla es grabándola en una cinta a cassette, utilizando para ello un magnetofón conectado al ordenador mediante el cable de doble clavija en cada extremo. También puede utilizar cintas grabadas y ejecutar sus programas. En el capítulo 8 de este manual, encontrará toda la información al respecto.

Una vez instalado su ordenador, deseará utilizarlo. El resto de este manual, le indicará cómo hacerlo, pero, probablemente llevado por su impaciencia, ya habrá pulsado algunas teclas y descubierto que se ha borrado el mensaje de arranque del sistema. No importa, **el ordenador no se puede estropear por eso.** Sea atrevido y experimente. Si no sabe como salir de esta situación, recuerde que puede inicializar el sistema pulsando el botón de Reset. Este debe ser siempre el último recurso, ya que al hacerlo perderá la información almacenada en el ordenador.

ADVERTENCIA: No utilice nunca periféricos que no estén concebidos para este ordenador.

2. El teclado

El teclado de su INVES SPECTRUM+ es muy similar al de una máquina de escribir convencional. Tanto los caracteres alfabéticos como los numéricos están colocados en la misma posición; sin embargo, las teclas pueden realizar más de una función. Por defecto, los caracteres se escriben en minúscula, pero cuando se utilizan en conjunción con la tecla (⇧) aparecen en mayúscula.

Para identificar el modo actual del teclado, en la pantalla se muestra una letra en inverso (blanco sobre fondo negro) que indica la posición del siguiente carácter a imprimir cuando se pulse una tecla. A esta letra intermitente en inverso se le denomina «cursor».

Al pulsar cualquier tecla aparecerá la palabra impresa encima de la letra en la tecla (llamada palabra clave). Esto es así porque el ordenador está esperando una orden de su parte para decirle lo que ha de hacer y todas las instrucciones deben comenzar con una palabra clave o comando. A diferencia de otros ordenadores, su INVES SPECTRUM+ le permite introducir palabras clave con sólo una pulsación de la tecla correspondiente.

Por ejemplo, si se pulsa la tecla **P** inmediatamente después del encendido, la palabra clave **PRINT** aparecerá en la pantalla.

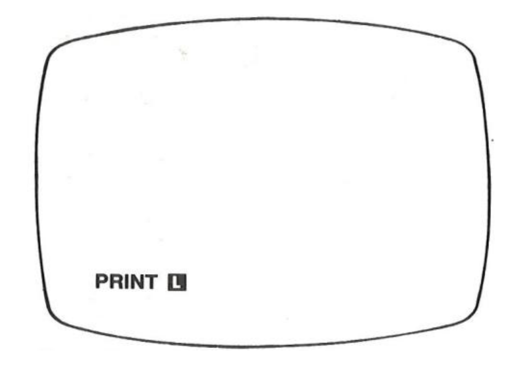
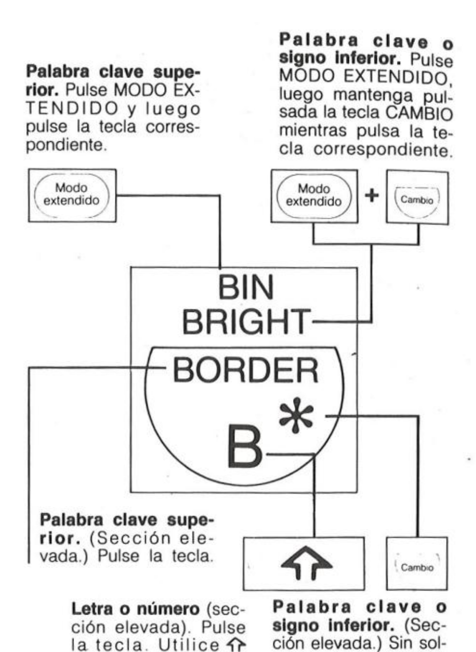


Figura 3

Ahora, el cursor cambia a una L que indica que el ordenador está esperando el dato que irá asociado con el comando PRINT introducido anteriormente. Teclee la palabra «Hola». Si hay algún otro texto en la pantalla, pulse el botón de Reset del ordenador y empiece de nuevo. Utilice la tecla ⇧ para obtener la mayúscula **H**. Para poder acceder a cualquier palabra,letra o número en su ordenador, tome nota primero de su posición sobre la tecla y luego utilice la misma secuencia de teclas selectoras que se indica a continuación:



Una sentencia que comience con **PRINT** comunica al ordenador que escriba en la pantalla las letras encerradas entre comillas. Para que el ordenador ejecute esta orden, debe utilizarse la tecla **ENTER**(↵). Cuando haya realizado esta operación, se verá en la pantalla la palabra

Hola

y algunos otros caracteres. (Una ? en inverso indica que hay un error en alguna parte.) Si ello sucede, empiece de nuevo y repita el ejercicio. El mensaje «OK», en la parte inferior de la pantalla, informa que todo se ha ejecutado correctamente. El mensaje es importante cuando se ejecutan programas, pero de momento podemos ignorarlo.

Hagamos alguna otra observación: La letra O y el número 0 se representan con caracteres diferentes. Es importante tenerlo muy en cuenta. El carácter numérico cero **siempre** tiene una línea cruzada.El ordenador siempre interpretará la letra O como un carácter alfabético, por lo que no ha de pulsarse la tecla

equivocada. Análogamente, el carácter numérico 1 y la letra minúscula L son diferentes y, a diferencia de algunas máquinas de escribir, no se pueden intercambiar.

Como es muy importante comprender los modos del teclado, será muy útil resumir lo que sucede una vez más.

El carácter L se denomina cursor. Indica en qué lugar de la pantalla colocará el cursor lo que Vd. vaya tecleando. No siempre es un carácter L si desconecta el ordenador y lo vuelve a encender o hace un reset y luego pulsa la tecla **ENTER**, el mensaje de arranque del sistema cambiará a K. La letra que aparece le comunicará cómo interpretará el ordenador lo que teclee a continuación. Al principio de una línea, será una K que significa «palabra clave» (el mensaje de arranque del sistema y los informes también se corresponden con el carácter K). Una palabra clave es una de las palabras especiales del ordenador, que figura al principio de una sentencia para dar al ordenador una idea de lo que quiere que haga. Puesto que el ordenador está esperando una palabra clave al principio de la línea, cuando pulse, por ejemplo, la tecla P el ordenador decide no interpretarla como una B sino como **PRINT** y le avisa de que así lo está haciendo al hacer aparecer el cursor en modo K. Cuando tiene la primera palabra clave, no espera otra, y por ello interpretará como datos todo cuanto teclee a continuación. Para poner de manifiesto tal circunstancia, el ordenador cambia el cursor a L (letras).

Estos estados diferentes suelen denominarse «modos» y hablaremos de **modo comando** (K) y de **modo letras** (L).

Si desea introducir muchas letras mayúsculas sin mantener pulsada la tecla ⇧, puede hacer que todas las letras aparezcan como mayúsculas pulsando primero ⇳ Para indicar que ello está sucediendo, el cursor se sustituirá por una C . Para obtener letras minúsculas y que el cursor vuelva a modo L, pulse la tecla ⇳ por segunda vez.

Además de las palabras clave, las letras, los números y diversas expresiones científicas y de programación, el teclado dispone también de ocho caracteres gráficos y sus correspondientes inversos. Estos están situados en las teclas de números de 1 a 8 y pueden imprimirse en la pantalla de una forma similar a la de las letras y números. Para hacerlo, el teclado debe cambiar a modo gráfico, lo que se realizará pulsando la tecla **GRAF**. Observe que el cursor cambia a G. Si vuelve a pulsar esta tecla volverá al modo L. Existe un último modo al que puede cambiar el teclado: El modo EXTENDIDO. Este modo se indica por un cursor E y se obtiene pulsando la tecla **MODO EXTENDIDO**. Ello permite el empleo de la mayor parte de las funciones científicas y de programación. Al pulsar de nuevo esta tecla, el teclado volverá al modo L. Aunque sea un mecanógrafo o programador muy cualificado, puede ocurrir que pulse teclas equivocadas. Hasta ahora el único método que conocía para subsanar el error era hacer un reset en el ordenador y empezar de nuevo. Aún cuando ello pueda ser adecuado si sólo se ha dado una orden al aparato, será muy inconveniente si se ha introducido ya mucha información. Afortunadamente, podemos utilizar la tecla **BORRAR** para eliminar los errores. Por ejemplo, poco puede salir mal con la simple orden:

PRINT «Hola»

.. o ¿puede haber algo erróneo?

Supongamos que no se acordó de poner las comillas al principio de la palabra. En la pantalla aparecería:

PRINT Hola»

El ordenador no hubiera reconocido lo que venía a continuación de **PRINT**, pues la no existencia de comillas indica al INVES SPECTRUM+ que se espera un número y, en su lugar, encuentra una letra. Mostrará su confusión mostrando un símbolo ? en el lugar del error.

Afortunadamente, no tenemos que teclearlo todo de nuevo. En la fila inferior del teclado hay cuatro flechas apuntando a distintas direcciones. Con las que están en posición horizontal, podrá desplazarse a izquierda o derecha hasta colocarse en el lugar donde faltan las comillas. Sitúe el cursor sobre la letra H y escriba las comillas que faltan. Compruebe que el resto de la instrucción está correcto.

NOTA: Una de las características de su INVES SPECTRUM+ que le será de gran utilidad en las correcciones, es la autorrepetición de cualquiera de las teclas. Si mantiene pulsada cualquier tecla más de tres segundos, se repetirá automáticamente a la vez que emite un sonido suave.

Ahora, cuando pulse **ENTER** (⏎) ordenador escribirá su mensaje en la parte superior de la pantalla (o debajo de otro ya existente y que sigue representado en la pantalla).

Podrá encontrar una descripción completa de todas las teclas en el último capítulo de este manual.

3. Números, letras y el ordenador como calculadora

Ya hemos visto cómo decirle al ordenador que imprima letras y caracteres gráficos en la pantalla, con el empleo del comando **PRINT.** También hemos visto que **ENTER** ha de emplearse para decirle al ordenador que ejecute la orden que se acaba de introducir por el teclado, Desde ahora en adelante, no utilizaremos **ENTER** en el manual cada vez que se utilice una orden o comando, sino que supondremos que la ha pulsado automáticamente al final de la línea.

Los números pueden manipularse por el ordenador más fácilmente que las letras. En el capítulo anterior aludimos a ello al explicar que el ordenador espera un número después de **PRINT**, si no se emplean comillas.

Por ello, si tecleamos:

PRINT 2

el número **2** aparecerá en la pantalla.

Es posible mezclar letras y números:

PRINT 2, «ABC»

Obsérvese que hay una separación o intervalo en la pantalla entre el 2 y ABC.

Ahora teclee:

PRINT2;«ABC››

y luego:

PRINT 2 «ABC››

Utilizando una coma entre los elementos después de **PRINT** se les espacia en 16 columnas, si se emplea un punto y coma (1) no se dejan espacios y si no se pone nada. Se produce un error.

PRINT puede utilizarse también con las funciones matemáticas del teclado De hecho, su INVES SPECTRUM+_ puede actuar como una calculadora electrónica.

Por ejemplo:

PRINT 2+2

La respuesta aparece en la parte superior de la pantalla. Compare esto con:

PRINT «2+2››

Es posible combinarlos para obtener algo más útil. Pruebe:

PRINT «2+2=››;2+2

Pruebe algunas otras expresiones aritméticas como:

PRINT 6-4 **PRINT 15/3** **PRINT 15*4**

El asterisco (*) se utiliza como signo de multiplicación en lugar de x para evitar confusiones con la letra **x**, y el símbolo de la barra inclinada (I) se emplea en la división en vez del signo /.

Experimente con diferentes tipos de cálculos. Si lo desea, puede utilizar números negativos o números con punto decimal.

Si llega a emplear las 22 líneas de la parte superior de la pantalla, observará que sucede algo interesante:Todose desplazará una línea hacia arriba y se perderá la línea superior. Esto se denomina «desplazamiento›› (scrolling).

Los cálculos no siempre se realizan en el orden que uno podría esperar. Pruebe por ejemplo:

PRINT 3+4*2

Podría esperar que la secuencia fuera: Tomar 3 y sumarle 4, lo que da como resultado 7; a continuación multiplicarle 2 a dicho resultado, obteniendo un total de 14. Sin embargo, esto no es así. Las multiplicaciones y las divisiones, también se realizan antes que las sumas y restas, por lo que la expresión «3+4*2›› significa «tomar 4 y multiplicarlo por 2, obteniendo 8 y luego sumarle 3, obteniendo 11». El número li debe ser la respuesta representada en la pantalla por el ordenador.

Dado que las multiplicaciones y las divisiones se realizan primero, decimos que tienen prioridad más alta que la suma y la resta. Entre si, la multiplicación y la división tienen la misma prioridad, lo que significa que las multiplicaciones y las divisiones se realizan en un orden de izquierda a derecha. Una vez efectuadas estas operaciones nos quedaran las sumas y las restas y como estas últimas tienen también la misma prioridad entre si, las realizaremos también en el orden de izquierda a derecha.

Veamos como el ordenador realizaría la operación siguiente:

PRINT 20-2*9+4/2*3

i 20-2*9+4/2*3 ii 20-18+4/2*3 iii 20-18+2*3	Primero realizamos las multiplicaciones y las divisiones de izquierda a derecha.
iv 20-18+6 v 2+6 vi 8	Luego las sumas y las restas.

Aunque todo lo que realmente necesita saber es si una operación tiene una prioridad más alta o más baja que la otra, el ordenador lo hace aplicando un número entre 1 y 16 para representar la prioridad de cada operación. Así * y / tienen prioridad 8 y + y - tienen prioridad 6.

Este orden de cálculo es absolutamente rigido, pero puede modificarlo empleando paréntesis; algo entre paréntesis se evalúa primero y luego se trata como un solo número, de modo que:

PRINT 3*2+2

da un resultado de 6+2=8, pero:

PRINT 3*(2+2)

será igual a 3*4=12.

A veces es útil dar al ordenador expresiones como estas porque cuando el ordenador esta esperando que le introduzca un número, puede proporcionarle, en cambio. Una expresión y le dará la respuesta correspondiente. Las excepciones a esta regla son tan pocas que se indicarán explícitamente en cada caso.

Puede escribir números con punto decimal (utilícese el punto ortográfico) y también puede emplear la notación científica, como es bastante frecuente en las calculadoras de bolsillo. En esta notación, después de un número ordinario (con o sin punto decimal) escribirá una parte exponencial constituida por la letra e, luego quizá - y a continuación un número. La parte exponencial desplaza el punto decimal hacia la derecha (o hacia la izquierda para un exponente negativo), con lo que se multiplica (o se divide) el número original por 10 unas cuantas veces. Por ejemplo:

2.34e0=2.34
2.34e3=2340
2.34e-2=0.0234 etc.

(pruebe a imprimir las expresiones en el ordenador). Este es uno de los pocos casos en que no puede sustituir un número por una expresión; por ejemplo no puede escribir:

(1.34+1)e(6/2)

También puede tener expresiones cuyos valores no sean números, sino cadenas de letras: Ha visto ya la forma más sencilla de esta cadena en muchas ocasiones y esta, es la cadena de letras escrita entre comillas. Es bastante análoga a la forma más simple de expresión numérica, que no es otra cosa que un número escrito sin ningún otro elemento. Lo que todavía no ha visto es el uso del signo más (+) con las cadenas (pero no los signos -, * ó /, por lo que no existe ningún problema de prioridades en este caso), Para sumar cadenas basta unirlas una a continuación de otra. Un ejemplo es:

PRINT «para››+«choques››

Puede «sumar» tantas cadenas como desee en una sola expresión y si así lo necesita, puede emplear incluso paréntesis.

4. Algunos comandos sencillos

La memoria de un ordenador puede utilizarse para almacenar toda clase de cosas. Hasta ahora, hemos visto como el comando PRINT nos permite ver en la pantalla letras, números y los resultados de cálculo con el empleo de letras y números.

Si necesitamos decirle al ordenador que recuerde un número o una cadena de letras, entonces, tenemos que asignar parte de la memoria para dicho uso.

La mayoría de las calculadoras de bolsillo tienen una tecla denominada «memoria››, que se utiliza para recordar números más tarde. Su ordenador puede hacer mucho más que eso. Puede tener tantas de esas casillas imaginarias de memoria como usted quiera y escribir un nombre en cada una.

Como ejemplo, suponga que necesita recordar su edad. El comando **LET** (comando situado en la tecla L) es el que se utiliza en este caso. Digamos que 34 es dicha edad.

LET edad= 34

Lo que sucede cuando se utiliza el comando LET es que una cierta sección de la memoria se designa « edad ›› y el número se almacena en ella. Para extraer esta información almacenada, teclee:

PRINT edad

y aparecerá el número 34. Es muy fácil cambiar el contenido de la «casilla» denominada «edad». Teclee:

LET edad=56

y luego:

PRINT edad

deberá aparecer 56 en la pantalla. «Edad›› es un ejemplo de una **variable** así llamada porque puede variar su valor. Es posible cambiar la impresión de un mensaje directamente en la pantalla y el valor de una variable. Teclee:

PRINT «su edad es ››; edad

Sin embargo, el ordenador es mucho más útil que simplemente recordar números cornombres unidos a ellos. También puede recodar cadenas de letras, Para diferenciar entre variables numéricas y variables de cadena (como se denomina), se emplea el símbolo del dólar ($) al final del nombre de la variable.

Por ejemplo, si necesitamos memorizar la cadena de letras:

«**Su edad es**»

podríamos llamarla (Los nombres de variables de cadena sólo pueden tener una sola letra, además del símbolo de $). Por ello teclee:

LET a$=«$u edad es »

Si ahora teclea:

PRINT a$

aparecerá la cadena de letras en la pantalla.

Si el ordenador no se ha desconectado desde el comienzo de este capítulo, teclee:

PRINT a$; edad

Y vea lo que sucede.

Hay otras formas de introducir información en la memoria del ordenador sin utilizar el comando **LET**.

Por ejemplo, el comando **INPUT**, en su forma más simple, comunica al ordenador que debe esperar alguna información procedente del teclado. En vez de teclear **LET**, etc., puede teclear cada vez:

INPUT edad

Una vez pulsada la tecla **ENTER**, un cursor intermitente aparecerá en la pantalla. E|lo significa que el ordenador necesita alguna informacion de su parte. En consecuencia, introduzca su edad y luego pulse la tecla **ENTER**. Aunque parezca que no ha sucedido nada, se le ha asignado a la variable el valor que usted ha introducido. Al teclear:

PRINT edad

se lo debe probar.

Combinemos todo ello con una serie de comandos.

Teclee:

LET b$=«¿Cuál es su edad?»
LET a5=« Su edad es »
INPUT (b$);edad:PRINT a$;edad

Observe que la última línea esta constituida por dos comandos separados por dos puntos (:)

INPUT (b$); edad

es otra forma de introducción de:

INPUT «¿cuál es su edad? » ;edad

5. Programación simple

Hasta ahora hemos estado diciendo al ordenador lo que ha de hacer directamente desde el teclado. Aunque es posible combinar comandos juntos, solamente son factibles aplicaciones limitadas con el empleo de este método.

Lo más importante respecto a los ordenadores es que son programables. Ello significa que podemos darles una serie de instrucciones para hacer que efectúen operaciones de forma secuencial.

Cada ordenador tiene su propio lenguaje que nos permite comunicar con el. Algunos lenguajes son muy sencillos, por lo que el ordenador puede comprenderlos con facilidad. Lamentablemente, lenguajes que son sencillos de comprensión para el ordenador no lo son tanto para los usuarios, En algunos aspectos, lo inverso también es cierto: Lenguajes que nos son bastante sencillos de comprender son relativamente difíciles para el ordenador e incluso han de traducirse o interpretarse.

Su INVES SPECTRUM+ utiliza un lenguaje de alto nivel denominado BASIC.

BASIC significa «Beginners All-purpose Simbolic Instruction Code» (Código de instrucciones simbólicas de uso general para principiantes) y fue concebido en la facultad universitaria de Dartmouth en new Hampshire, Estados Unidos, en 1964. Es muy ampliamente utilizado en los ordenadores personales, pero aunque es muy semejante en todos ellos hay sutiles diferencias, Esa es la razón por la que este manual se elabora concretamente para su INVES SPECTRUM+. Pero el BASIC del INVES SPECTRUM+ no tiene diferencias muy notables con el resto de las versiones de BASIC existentes, por ello, no debe tener demasiadas dificultades en adaptar cualquier programa en BASIC para su ejecución en el INVES SPECTRUM+. A diferencia de otras versiones, la de su ordenador no permite que se omita el comando **LET** cuando se asignan valores a las variables.

Hay un límite para el número de instrucciones que pueden almacenarse en el ordenador. Su INVES SPECTRUM+ indica este límite emitiendo un zumbido.

Cuando se programa en BASIC es necesario hacer conocer al ordenador el orden en que han de ejecutarse las instrucciones. De aquí que cada línea de la secuencia de instrucciones tenga un número en su comienzo. Es normal empezar en 10 e ir incrementando en 10 cada nueva línea. Esto permite que se inserten otras líneas si se hubieran omitido, o fuera necesario modificar el programa.

Examinemos un programa sencillo. Consideremos la serie de comandos del último capítulo. Si necesitásemos repetir la serie de comandos, sería necesario introducirlos cada vez. Un programa resuelve el problema eliminando dicha necesidad.

Introduzca lo siguiente con **ENTER** después de cada línea.

```
10 LET B$=«¿Cuál es su edad?››
20 LET As=«Su edad es»
30 INPUT(B$);edad
40 PRINTA$;edad
```

Observe que no es necesario introducir ningún espacio, salvo entre comillas. Nada sucedera realmente hasta que digamos al ordenador que comience a ejecutar el programa. Esta orden se comunica con el empleo de **RUN** (comando de la tecla **R**).

Introduzca este comando y vea lo que sucede.

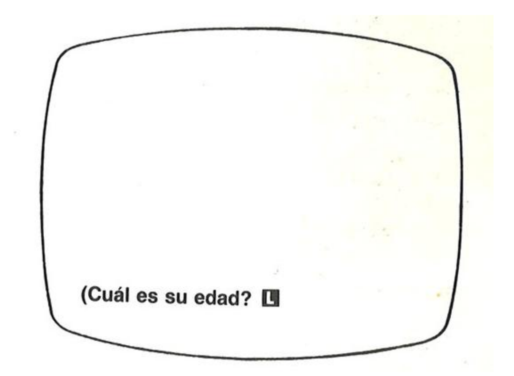


Figura 4

También puede haber notado una flecha apuntando a la derecha cuando se ha introducido cada línea. Ella indica la última línea introducida. Si necesita volver a ver el programa, pulse **ENTER** (o **LIST**). Puede emplear **RUN** para ejecutar el programa tantas veces como quiera. Cuando ya no necesite el programa, puede suprimirlo con el empleo del comando **NEW**, el cual borra el programa almacenado en memoria y le proporciona una «pizarra limpia» dispuesta para que escriba uno nuevo.

Teclee **NEW**, luego **ENTER** y vea lo que sucede.



Figura 5

Para recapitular:
Cuando teclee un comando precedido por un número, éste comunicará al ordenador que no se trata simplemente de un comando, sino de una línea de programa. El ordenador no procede a ejecutar la instrucción correspondiente, sino que realiza su almacenamiento para más tarde.
Su INVES SPECTRUM+ escribe en la pantalla (o lista) todas las líneas de programa que ha introducido con un ⊵ en la ultima línea.
El ordenador no ejecutará inmediatamente ninguna de estas líneas, sino que las va almacenando en su interior.
Para conseguir que el ordenador ejecute estas líneas, debe emplear el comando **RUN**.
Si pulsa **ENTER**, conseguirá un nuevo listado.
Consideremos otro programa sencillo. Este será algo más matemático e imprimirá los cuadrados de todos los números entre 1 y 10 (el cuadrado de un número es dicho numero multiplicado por si mismo).
Para generar números de 1 a 10 se introduce otro concepto en la programación en BASIC. Se trata del método mediante el cual conseguimos que cuente el ordenador. Anteriormente vimos que los números pueden almacenarse en la memoria del ordenador Uniendo un «nombre» a los mismos, o dicho técnicamente, asignando un valor a una variable. Sea la variable x que comienza con el valor 1 y que ha de incrementarse en pasos de 1 hasta 10. Esta operación se realizará con el empleo del comando **FOR… TO … STEP**.
Por tanto, para introducir este programa, teclee **NEW** para deshacerse del anterior y escriba lo siguiente:

10 FOR x=1 TO 10 STEP 1

(la parte STEP 1 normalmente puede omitirse si el contaje se incrementa en pasos de t).
La siguiente línea debe comunicar al ordenador lo que ha de hacer con x para cualquier valor que tenga. Para ello teclee:

20 PRINT x,x*x

Finalmente, necesitamos una linea para decirle al ordenador que vaya al siguiente valor de x; por consiguiente, teclee:

30 NEXT x

Al llegar a esta instrucción, el ordenador vuelve a la línea 10 y repite la secuencia. Cuando **x** supera a 10, el ordenador pasa a la siguiente línea en el programa, esto es, la línea 40.
El programa debe aparecer ahora en la pantalla como sigue:

10 FOR X=1 TO 10 STEP 1
20 PRINT X,X*X
30 NEXT X

Para la conclusión, debemos tener otra línea que comunique al ordenador que el programa ha finalizado cuando x=10; por tanto, teclee:

40 STOP

Si el programa está ahora en RUN (ejecución) deben aparecer dos columnas, la primera con los valores de x y la segunda con los valores de x*x ó x al cuadrado Es posible etiquetar estas columnas añadiendo otra línea, tal como:

5 PRINT «x››,«x*x››

Nótese que aunque esta línea se haya introducido *después* de todas las demás líneas, por tener un número de línea más bajo el ordenador la coloca automáticamente en el lugar correcto.
Intente escribir programas con el empleo de otras funciones matemáticas. Si tiene alguna duda sobre como utilizarlas, diríjase al capitulo dedicado a los comandos de BASIC.

6. Utilización del magnetofón a cassette

Resulta bastante monótono tener que teclear los programas para el ordenador cada vez que quiera utilizarlo. Su INVES SPECTRUM+ dispone de medios para la grabación de programas en cinta magnética con un magnetofón a cassette convencional. Si tiene un programa en memoria, pruebe a conservarlo siguiendo el procedimiento siguiente.

Si puede conservar programas en cinta a cassette los podrá cargar de nuevo más adeante.

Servirán la mayoría de los magnetofones existentes en el mercado. En lo que respecta al ordenador, los magnetofones monofónicos portátiles de bajo costo son al menos tan buenos como los estéreos más caros y plantearán también menos problemas. Será de gran utilidad que el magnetofón disponga de contador de cinta.

El magnetofón debe tener una conexión de entrada para micrófono y una conexión de salida para auriculares (si no la hubiere, pruebe con el enchufe del altavoz exterior). Deben ser conexiones hembra de 3,5 mm. (esto es para admitir adecuadamente los enchufes de clavija existentes en los cables suministrados), porque otras clases no suelen dar una señal suficientemente potente para el ordenador.

Podrá servir cualquier cinta a cassette aunque son preferibles as de bajo nivel de ruido. Una vez que disponga del magnetofón adecuado, debe conectarlo al ordenador con el empleo del cable doble suministrado con su INVES SPECTRUM+. Uno de los cables debe conectar el enchufe de entrada de micrófono del magnetofón, con el conector marcado «MIC›› de la parte posterior del ordenador y el otro, el enchufe de salida de los auriculares del magnetofón, con el conector «EAR›› del ordenador. (No puede dañar el ordenador conectando incorrectamente los cables.)

Cuando este utilizando el comando **SAVE** para almacenar un programa en cinta, no es necesario que desconecte ninguna clavija del cable que conecta las tomas de auricular del ordenador y el magnetofón, ya que no se producirá ningún tipo de interferencia y la grabacion se realizará correctamente, además de evitarle escuchar el zumbido, un tanto molesto que produce esta operación. Sólo debe tener en cuenta que, cuando llegue el momento de utilizar un programa que incluya sonido, deberá desconectar la clavija MIC para no perder dicho sonido.

Introduzca algún programa en el ordenador, por ejemplo el programa de los cuadrados considerado en el capítulo anterior, y teclee:

SAVE «cuadrados»

Cuadrados no es otra cosa sino el nombre que ha usado para etiquetar el programa de los cuadrados, mientras esta en la cinta. Le están permitidos hasta diez caracteres en el nombre, el cual debe estar constituido exclusivamente por letras y números.

El ordenador habrá presentado el mensaje **PREPARE LA CINTA Y PULSE ENTER**. Primero realizaremos un funcionamiento de prueba para que pueda ver lo que sucede. No ponga en marcha el magnetofón, sólo pulse una tecla en su INVES SPECTHUM+ y observe el contorno de la pantalla de TV. Verá configuraciones de bandas horizontales coloreadas.

-- Cinco segundos de bandas rojas y azul claro de 1 cm. de anchura, desplazándose lentamente hacia arriba.

-- Una ráfaga muy corta de bandas azules y amarillas (un segundo es la duración normal).

-- Dos segundos con la configuración de bandas rojas y azul claro otra vez.

-- Y nuevamente la configuración azul y amarilla hasta el final del proceso de carga.

Inténtelo de nuevo para que pueda reconocer todo lo indicado. La información se conserva en dos bloques y ambos tienen una parte de entrada inicial que corresponde a la configuración roja y azul claro y la información propiamente dicha, que corresponde a la configuración azul y amarilla. El primer bloque es de tipo preliminar y contiene el nombre y otros bits de información acerca del programa, mientras que el segundo bloque es el programa propiamente dicho junto con cualquier variable existente. La sección blanca entre ellos es un intervalo de separación.

Procedamos ahora a la grabación de la señal de la cinta.

1. Ajuste la posición de la cinta en una zona que este virgen o que este dispuesto a regrabar.
2. Teclee:

SAVE «cuadrados›› (y ENTER)

3. Inicie la grabación.
4. Pulse cualquier tecla de su INVES SPECTRUM+.
5. Observe la pantalla de televisión como antes. Cuando haya acabado el ordenador (con la aparición del informe **0 OK**), inte-rrumpa el funcionamiento del magnetofón.

Para cerciorarse de que la grabación se ha producido, puede comprobar la señal en la cinta con respecto al programa en el ordenador con el empleo del comando **VERIFY**.

1. Gire el mando de volumen del magnetofón a aproximadamente la mitad de su recorrido y vuelva a conectar el cable de «**EAR**».

2. Rebobine la cinta hasta el punto anterior a donde inició la grabación momentos antes.

3. Teclee:

VERÍFY «cuadrados»

(**VERIFY** es un comando del modo extendido y por tanto, accederá a él pulsando la tecla R en dicho modo.)

4. Comience la reproducción en el magnetofón.

El contorno de la pantalla de televisión alternará entre rojo y azul claro hasta que la cinta llegue a la grabación que ha realizado; entonces vera la misma configuración que cuando realizó la grabación del programa. En el intervalo de un segundo se escribirá en la pantalla **Programa: cuadrados** (cuando el ordenador esté buscando algo en la cinta, imprimirá el nombre correspondiente). Si observa toda esta configuración y luego se interrumpe el funcionamiento del ordenador con el informe **0 OK**, ello indica que su programa esta seguramente grabado en la cinta y puede pasar a los siguientes párrafos. De no ser así, algo se hizo erróneamente. Pase a las cuestiones siguientes para averiguar qué fue lo que ocurrió.

Cómo asegurarse de que su programa está grabado

¿Ha aparecido en la pantalla el nombre del programa?

Si la respuesta es negativa, el programa no se grabó adecuadamente o si lo hizo no fue objeto de una adecuada lectura posterior. Necesita averiguar la razón de la anomalía. Para comprobar si se guardo adecuadamente, rebobine la cinta hasta un poco antes de donde comenzó la grabación y efectúe la reproducción a través del propio altavoz del magnetofón (probablemente tenga que desenchufar el cable de conexión de los auriculares en el magnetofón). La entrada correspondiente a la configuración roja y azul claro da una nota permanente de alto tono y muy nítida y la información de la parte azul y amarilla da un sonido mucho menos agradable, como un mensaje de código Morse en un huracán. Ambos sonidos son bastante fuertes y, a pleno volumen) fácilmente pueden ahogar una conversación.

Si no oye estos ruidos, es probable que el programa no se haya conseguido guardar. Compruebe que los cables están enchutados en los receptáculos correctos Cerciórese de que los enchufes «MIC›- están conectados. Con algunos magnetofones sucede que las clavijas no nacen buen contacto si se introducen a fondo. Pruebe a sacarlas una décima de pulgada aproximadamente; en ocasiones podrá notar como se asientan en una posición mas natural Compruebe también que no haya estado intentando grabar en el principio de la cassette, donde la cinta carece todavía del baño magnético. Una vez comprobado todo esto, intente efectuar nuevamente la grabación del programa.

Si puede oír los sonidos antes indicados, es probable que haya realizado adecuadamente la función de grabación **SAVE** y que su problema radique en la lectura.

Vuelva a comprobar los cables y también el nivel de volumen. Si es pequeño, el ordenador no oirá adecuadamente la señal y Vd. no verá las configuraciones correctas en la pantalla; si fuera demasiado fuerte, la señal se distorsionará. Hay una amplia gama de valores intermedios aceptables. y por tanto, lo mejor es que experimente con ellos.

El siguiente caso es cuando el ordenador encuentra el programa y escribe su nombre, pero todavía esta equivocado. Algunas posibilidades son:

Un error de sintaxis en el nombre, bien sea en **SAVE** (en cuyo caso el ordenador escribirá el nombre incorrectamente en la pantalla) o en **VERIFY** (el ordenador ignorara el programa y aparecerá la configuración alternada de colores rojo y azul claro).

Hay un error típico en la cinta y es cuando el ordenador presenta el mensaje: **R ERROR DE CARGA**, que significa en este caso que dejo de verificar el programa. Habrá de efectuar nuevamente la operación de grabación del programa con el comando **SAVE**.

También es posible que el ajuste de volumen en el magnetofón no sea correcto, pero no puede ser muy inadecuado porque el ordenador trato de efectuar la lectura del primer bloque.

Supongamos ahora que ha conservado el programa y lo ha verificado satisfactoriamente. El volverlo a cargar es prácticamente como verificarlo, con la salvedad de que debe escribir:

LOAD «cuadrados»

en lugar de:

VERIFY «cuadrados»

La función **LOAD** esta ubicada en la tecla **J**. Puesto que la verificación fue satisfactoria, no debe tener ningún problema con la carga.

LOAD borra el programa antiguo (y las variables) en el ordenador, antes de cargar el nuevo de la cinta.

Una vez que se haya cargado el programa, lo ejecutará el comando **RUN**.

Si su cinta tiene mas de un programa grabado en la misma cara, entonces cada uno tendrá un nombre que lo identifique. Puede elegir que programa cargar con el comando **LOAD**; por ejemplo, si el que desea cargar se denomina «geografia››, podría teclear:

LOAD «geografia»

(**LOAD** “” significa CARGAR el primer programa que encuentre, lo cual puede ser de gran utilidad si no puede recordar el nombre de su programa.)

7. Colores

Una de las buenas razones para adquirir el ordenador INVES SPECTRUM+ es en primer lugar la posibilidad de utilizar color en la pantalla de TV. La pantalla se divide en dos zonas. La parte exterior se denomina **BORDER** (contorno) y la zona central **PAPER** (papel). Es posible cambiar los colores de estas dos secciones a voluntad, tanto directamente por medio del teclado como por programa.

El INVES SPECTRUM+ tiene una gama de ocho colores, identificados con números de 0 a 7. En un televisor de blanco y negro aparecerán como tonalidades de gris.

He aquí la lista de los colores con los numeros asociados a los mismos:

0 negro
1 azul
2 rojo
3 púrpura o magenta
4 verde
5 azul pálido o cyan
6 amarillo
7 blanco

Cuando el ordenador se conecta por primera vez, el sistema trabaja en blanco y negro. Por ello, el valor por defecto para **BORDER** y **PAPER** es 7, es decir, blanco. El color de cualquier carácter que aparece en la pantalla se define por el comando **INK** (tinta). Este suele ser 0, o sea, negro. Inicialmente, los tres comandos que controlan los colores de la pantalla están ajustados por el ordenador.

Sin embargo, Vd, puede cambiar estos valores. Por ejemplo, teclee:

BORDER 2

Si recordó pulsar la tecla **ENTER**, el contorno deberá cambiar ahora de blanco a rojo. Se incluye la zona de la parte inferior en donde se introducen los comandos e instrucciones. Pruebe a teclear otros números y verá como cambian los colores.

Intente cambiar la zona central de la pantalla, tecleando:

PAPER 5

El comando **PAPER** es uno de los del modo extendido, según se menciono anteriormente y se encuentra en la tecla **C**. Una vez introducido el comando, si se pulsa la tecla **ENTER** dos veces, la zona central de la pantalla debe cambiar a azul pálido. La primera pulsación de **ENTER** cancela el comando **PAPER** ya almacenado en el ordenador, pero sólo cuando se pulse **ENTER** por segunda vez (con lo que se hace que el ordenador efectúe el LISTADO de un programa y, por tanto, reconstruya la información de la pantalla) se consigue el empleo del nuevo color de **PAPER**. Si se está utilizando un televisor color y éste no ha cambiado, pruebe a_ ajustar los mandos del color del aparato y quizá el mando de sintonía.

El comando INK es semejante al comando **PAPER** y controla el color de los caracteres que aparecen en la sección **PAPER** de la pantalla. Evidentemente, si los colores de **INK** y de **PAPER** son los mismos no aparecerá nada en la pantalla.

Los comandos **BORDER**, **PAPER** e **INK** pueden emplearse en programas. A continuación damos uno sencillo para mostrar la gama de colores de que dispone.

```
10 FOR X=0 TO 7
20 BORDER X
30 PAPER 7-X:CLS
40 PAUSE 50
50 NEXT X
```

Este programa, cuando está en ejecución **(RUN)**, pasa a través de los ocho colores, contrastando los colores de **PAPER** y de **BORDER**. El comando **CLS**, después de **PAPER** obliga al ordenador a reconstruir la imagen de la pantalla y a emplear el nuevo color de **PAPER**. El comando PAUSE interrumpe el programa durante un segundo, para que podamos ver lo que está sucediendo (pruebe a ejecutar el programa omitiendo el comando **PAUSE**). Para mostrar como actúa el comando **INK**, introduzca por teclado el siguiente programa, después de un comando NEW.

```
10 BORDER 7
20 PAPER 1
30 INK 4
40 PRINT «Caracteres verdes sobre fondo
azul ››
```

Hay otros comandos asociados con las capacidades de colores del INVES SPECTRUM+ que se detallaran en el último capítulo.

8. Sonido

El ordenador INVES SPECTRUM+ puede producir sonidos de una variedad prácticamente ilimitada. La frecuencia de la nota y su duración están bajo el control del usuario. El comando **BEEP** se utiliza para comunicar al ordenador que, emita un sonido. **BEEP** es un comando del modo extendido y se encuentra en la tecla **Z**.

La frecuencia «central» para el comando **BEEP** es el «DO» central (C central). Esta frecuencia puede variarse dentro del comando **BEEP** y puede obtenerse cualquier nota si se expresa como semitonos o partes de semitonos por encima o por debajo de dicha frecuencia central. Si el comando

```
BEEP 2,0
```

se introduce, el ordenador debe emitir un sonido de un tono de «do» (C) central durante dos segundos.

Los dos números dados controlan la clase de nota que se emite, el primero da una duración de la nota en segundos y el segundo la altura de la nota en semitonos por encima del -do» central. Así, el código de altura del so-nido para «do» central es 0, el del «do» sos-tenido (C=) es 1, el del «re» (D) es 2 y así sucesivamente, hasta el siguiente «do» (C) de una octava superior que es 12, porque 12 semitonos constituyen una octava. Puede continuar con 13 y más allá si así lo desea, de modo que cuanto más alto sea el número, tanto más alto será el tono del sonido.

Pruebe a introducir:

```
BEEP 1,4:BEEP 1,2:BEEP 2,0
```

Debe oír al menos el primer compás de una conocida melodía. Puesto que puede unir bastantes BEEPs juntos con dos puntos (:) como en la expresión anterior, podría, si tuviera la paciencia necesaria, obtener un tono completo. Le podría atraer el empleo de más de tres notas.

(Los dos puntos no sólo unen **BEEP**s; puede utilizarlos para elaborar comandos compuestos a partir de algunos de los comandos elementales.)

A título de ejemplo más complicado, puede elaborar una instrucción, mezclando varios comandos **BEEP** y **BORDER** juntos:

```
BORDER 1:BEEP 1,14:BORDER 3:BEEP 1,
16:BORDER 4:BEEP 1,12: BORDER 6:BEEP
1, 0:BORDER 5:BEEP 4, 7:BORDER 1
```

(No se preocupe por el hecho de que se extienda de una línea a la siguiente; el ordenador no se entera de tal circunstancia.)

Un breve programa para tocar una serie completa de notas podría ser el siguiente:

```
FOR x=0 TO 24
20 BEEP 2,x
30 NEXT x
```

Hay muchas más cosas que pueden hacerse con este comando; vea el último capítulo de este manual para tener una idea más clara sobre la programación en BASIC.

Para las notas por debajo del «do» central, el número de semitonos se indica por un número negativo.

9. ¿Cómo es el interior del ordenador INVES SPECTRUM+?

En la ilustración adjunta se muestra cómo es el interior del ordenador INVES SPECTRUM+.

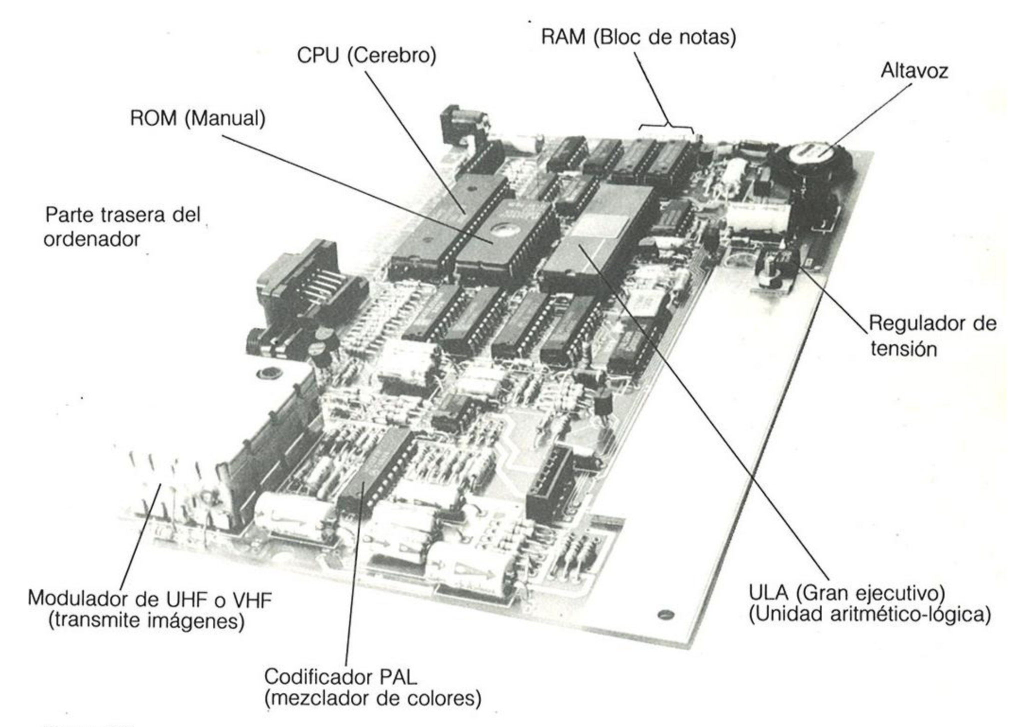


Figura 10

Como puede observar, todo, o casi todo, se nombra por una abreviatura de tres letras. Los elementos rectangulares negros de plástico, con patillas metálicas, son los circuitos integrados que realmente hacen todo el trabajo. En el interior de cada uno hay un elemento cuadrado de 1/4 x 1/4 de pulgada de silicio unido por hilos de conexión a las patillas metálicas. En esa pastilla de silicio hay miles de transistores que constituyen los circuitos electrónicos que son el ordenador.

El cerebro que hay detrás de todo este complejo es la pastilla integrada del procesador, denominada CPU (Unidad Central de Proceso). En este caso particular, la CPU es una Z80A, que es una versión más rápida del popular Z80 de Zilog.

El procesador controla el ordenador, realiza los cálculos aritméticos, examina qué teclas se han pulsado, decide lo que hay que hacer como consecuencia y, en general, decide lo que el ordenador debe realizar en cada momento. Sin embargo, a pesar de su inteligencia, no podría hacer todo lo anterior por su propia cuenta. No sabe nada acerca de BASIC, ni sobre la aritmética de coma decimal, por ejemplo, y ha de obtener todas las instrucciones de otra pastilla integrada, la ROM (Memoria de Sólo Lectura). La ROM contiene una larga lista de instrucciones que forman un programa de ordenador y que dice al procesador lo que ha de hacer en todas las circunstancias previsibles. Este programa no está escrito en BASIC, sino en lo que se denomina Có-

digo-Máquina del Z80 y adopta la forma de una larga secuencia de números. Hay un total 16384 (16*1024) y ésta es la razón de que al BASIC del INVES SPECTRUM+ se le denomine, a veces, un 16K BASIC (1K equivale a 1024 bytes).

Aunque hay pastillas integradas similares en otros ordenadores, esta secuencia particular de instrucciones es exclusiva para el INVES SPECTRUM+ y se escribió especialmente para el mismo.

Las dos pastillas integradas próximas a la ULA son para la memoria. Esta es la RAM (Memoria de Acceso Aleatorio) y hay otras dos pastillas que actúan de forma muy vinculada a las anteriores. La RAM es en donde el ordenador almacena la información que precisa conservar, algunos programas en BASIC, las variables, la imagen para la pantalla de televisión y diversos otros elementos que mantienen el registro del estado del ordenador.

La pastilla grande es la ULA (Unidad Aritmético-Lógica). Actúa realmente como el «centro de comunicaciones», que se cerciora de que todo lo que requiere el procesador se realiza oportunamente. Asimismo, lee la memoria para ver de qué consta la imagen de televisión y envía las señales adecuadas al dispositivo de interconexión («interfase») de TV.

El codificador PAL es un grupo completo de componentes que adapta las señales de video de su ordenador a una salida adecuada a los sistemas de T.V. tipo PAL.

El regulador convierte la tensión algo errática de la fuente de alimentación en una tensión de 5 voltios absolutamente constante.

10. El BASIC de su ordenador INVES SPECTRUM +

Este capítulo describe los detalles del BASIC que emplea su INVES SPECTRUM+ es un sumario sobre su empleo y otros detalles adicionales de las palabras clave. Esta información cubre los aspectos necesarios para crear desde programas sencillos hasta los más complejos y avanzados. Este capítulo no se ha organizado para su lectura desde el co-mienzo hasta el final. Su estructura es más bien la de un diccionario de programación, y le permitirá sacar el máximo provecho de su INVES SPECTRUM+.

Guía de referencia del programador; uso de las palabras clave del BASIC del INVES SPECTRUM+.

CLASES DE PALABRAS CLAVE

Las palabras clave pueden clasificarse en-tre una y cuatro clases.

Comandos

Un comando es una palabra clave que le in-dica al ordenador que debe ejecutar una determinada acción. Ciertos comandos se pondrán en práctica inmediatamente, por ejemplo: RUN, LOAD.

Instrucciones

Como los comandos, las instrucciones son palabras clave que le indican al ordenador que debe ejecutar una determinada acción. La diferencia entre ambos radica en que las instrucciones están incluidas en un programa, y por lo tanto se llevan a la práctica únicamente cuando el ordenador ejecuta el programa.

Ejemplo DRAW, INPUT.

Función

Es una palabra clave que produce un valor. La función puede formar parte de un comando o una instrucción. Ejemplos: RND, INT.

Operador lógico

Es una palabra clave que se emplea para realizar procesos lógicos.

Puede incluirse en un comando o en una instrucción y determina si una condición especificada es verdadera o falsa.

Esta guía contiene una descripción com-pleta de todas las palabras clave en BASIC disponibles en el INVES SPECTRUM+. Cada entrada incluye información sobre los siguientes aspectos:

- Posición de la palabra clave.
- Clase de palabra clave.
- Propósito de la utilización de esta palabra clave.
- Empleo de la palabra clave.
- Formato utilizado al redactar un programa que contiene la palabra clave.
- Código del conjunto de caracteres.

Los detalles referentes a la posición, propósito, empleo y código del juego de caracteres no requieren más explicación. La información sobre clase y formato es algo más completa. Por lo tanto, le sugerimos que, para hacer un mejor uso de la guía, primero estudie cuidadosamente la información en esta página.

NUMEROS Y VARIABLES

Números

Se almacenan con una precisión de 9 a 10 dígitos. La gama de números que se pueden manipular va desde 10^{38} hasta $4*10^{-39}$.

Variables aceptadas

Numéricas. De cualquier longitud, comenzando por una letra. Se ignoran los espacios y todas las letras se convierten en minúsculas, por tanto no se distingue entre mayúsculas y minúsculas.

Cadena. Cualquier letra única seguida por un signo de dólar, $. No se distingue entre mayúsculas y minúsculas.

Matriz. Para variables de matriz, véase la sección que trata sobre DIM.

FORMATO DE LAS PALABRAS CLAVE

La sección que trata del formato de las palabras clave describe la sintaxis correcta de cada palabra clave, y demás factores in-cluidos en una instrucción que se ha introd.-cido al ordenador. Los elementos que se com-binan con palabras claves incluyen variables y valores. En el texto para describir el formato correcto de una determinada palabra clave, utilizaremos las siguientes abreviaciones

Abreviación	Explicación	Ejemplo
const-num	Una constante numérica (número)	24,5
var-num	Una variable numérica (una variable que puede contener un número).	suma
expr-num	Expresión numérica (cualquier combinación de constantes y variables numéricas que producen un número como resultado).	Suma*24.5 RND*7
const número entero var número entero expres número entero	Una constante, variable o expresión numérica cuyo valor se redondea al número más próximo	
const de cadena	Una constante de cadena o cadena (cualquier combinación de carácteres entre comillas)	«Zx spectrum+»
var de cadena	Una variable de cadena (una variable que puede contener una cadena)	a$
exp. de cadena	Una expresión de cadena (cualquier combinación válida de constantes de cadena y variables de cadena que suministre una cadena)	a$+«INVES SPECTRUM+» a$(6 TO 8)
letra	Cualquier letra mayúscula o minúscula	Y x
letras$	Cualquier letra mayúscula o minúscula seguida del sigilo $	B$ a$
cond	Una subcondición, o una sub-condición insertada dentro de una condición.	x=10 **AND** t<10 condición
instrucción [ ]	Cualquier instrucción en BASIC válida al ser utilizada. Un elemento opcional que puede ser repetido	**IF** t>10 **THEN STOP**

NOTA: En el texto los términos valor numérico (numeric value) y valor de cadena (string value) denotan cualquier elemento numérico o de cadena respectivamente.

LOS SIGNOS DEL BASIC DE INVES SPECTRUM +

Signo	Ubicación	Acción/Use	Signo	Ubicación	Acción/Use
$	4	Variable de cadena	+	K	Suma. Número positivo
'	7	Inicia nueva linea	=	L	Igual
(	8	Abre paréntesis	:	Z	Separa instrucciones en un programa
)	9	Cierra paréntesis	/	V	División
<=	Q	Menor o igual que	*	B	Multiplicación
<>	W	Diferente de	.	Tecla .	Punto decimal
>=	E	Mayor o igual que	;	Tecla ;	Controla visualización de información en la pantalla. Separa instrucciones dentro de una misma línea de programa
<	R	Menor que	»	Tecla »	Abre y cierra una cadena
>	T	Mayor que	,	Tecla ,	Controla visualización de información en la pantalla. Ubica información en las columnas Ø o 16 en la pantalla. Separa a los valores colocados a continuación de una palabra clave.
↑	H	Elevar a la potencia			
–	J	Resta. Número negativo			

JUEGO DE CARACTERES DEL INVES SPECTRUM +

	0	1	2	3	4	5	6	7	8	9
0							**PRINT** coma	**EDITAR** **PAPER**	cursor izquierda	graficos derecha
10	cursor abajo	cursor arriba	**BORRAR**	⏎	número		**INK** control	control	**FLASH** control	**BRIGHT** control
20	**VIDEO** control	**OVER** control	**AT** control	**TAB** control						
30			espacio	!	"	#	$	%	&	'
40	(	)	*	+	,	–	.	/	Ø	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	/	]	↑	_	£	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~	©	□	▝
130	▘	▀	▗	▐	▚	▜	▖	▞	▌	▛
140	▄	▟	▙	█	GRAFICOS A	GRAFICOS B	GRAFICOS C	GRAFICOS D	GRAFICOS E	GRAFICOS F
150	GRAFICOS G	GRAFICOS H	GRAFICOS I	GRAFICOS J	GRAFICOS K	GRAFICOS I	GRAFICOS M	GRAFICOS N	GRAFICOS O	GRAFICOS P
160	GRAFICOS Q	GRAFICOS R	GRAFICOS S	GRAFICOS T	GRAFICOS U	RND	INKEYS	PI	FN	POINT
170	SCREENS	ATTR	AT	TAB	VALS	CODE	VAL	LEN	SIN	COS
180	TAN	ASN	ACS	ATN	LN	EXP	INT	SQR	SGN	ABS
190	PEEK	IN	USR	STRS	CHRS	NOT	BIN	OR	AND	<=
200	>=	<>	LINE	THEN	TO	STEP	DEF FN	CAT	FORMAT	MOVE
210	ERASE	OPEN #	CLOSE #	MERGE	VERIFY	BEEP	CIRCLE	INK	PAPER	FLASH
220	BRIGHT	INVERSE	OVER	OUT	LPRINT	LLIST	STOP	READ	DATA	RESTORE
230	NEW	BORDER	CON-TINUE	DIM	REM	FOR	GO TO	GO SUB	INPUT	LOAD
240	LIST	LET	PAUSE	NEXT	POKE	PRINT	PLOT	RUN	SAVE	RANDOM-IZE
250	IF	CLS	DRAW	CLEAR	RETURN	COPY				

ABS ABSolute value Valor Absoluto

Posición en el teclado

MODO EXTENDIDO
G

Función

La palabra clave ABS produce la magnitud absoluta de un valor numérico. Esto es, el valor sin un signo de positivo o negativo.

Cómo emplear ABS

ABS va seguido de un valor numérico. Si es una expresión debe ir encerrada .entre paréntesis. Por ejemplo:

50 LET x = ABS(y—z)

ABS retorna el valor absoluto del valor numérico.

Ejemplo

El comando

PRINT ABS-34,2

produce como resultado que el ordenador visualice el número 34,2.

Formato

ABS const-num
ABS var-num
ABS (expr-num)

ACS Arc CoSine Arco CoSeno

Posición en el teclado

MODO EXTENDIDO
CAMBIO W

Función

ACS calcula el valor de un ángulo a partir de su coseno.

Cómo emplear ACS

ACS debe ir seguido de un valor numérico. Si es una expresión, ésta debe estar entre paréntesis. Por ejemplo:

60 **LET x** = ACS(y*z)

El valor a continuación de la palabra clave ACS (y z en el ejemplo) es el coseno del ángulo requerido y puede variar entre + 1 y — 1. ACS retorna el valor del ángulo en radianes (no en grados). Para convertir los radianes en grados se multiplica el valor retornado por ACS por 180/PI.

Ejemplo

PRINT 180/PI* **ACS** 0.5

El resultado es 60, ángulo en grados que corresponde a un coseno de valor 0.5.

Formato

ACS const-num
ACS var-num
ACS (expr-num)

AND

Posición en el teclado

CAMBIO Y

Operador lógico/función

AND opera como un operador lógico, para determinar la veracidad de una combinación de condiciones. Solamente produce el efecto deseado si todas las condiciones enumeradas en la sentencia son verdaderas. AND también actúa como una función para ejecutar operaciones binarias en dos valores de cadena o dos valores numéricos.

Cómo emplear AND

Como un operador lógico. AND vincula dos condiciones incluidas en una instrucción en la que se desea establecer la veracidad del conjunto. Por ejemplo:

90 **IF** x=y+z **AND** tiempo<10 **THEN -PRINT** «Correcto»

Unicamente si ambas condiciones (x = y + z) y (tiempo >10) son verdaderas, el ordenador visualizará la palabra Correcto en la pantalla. Si cualquiera de las condiciones es falsa, la totalidad de la instrucción es falsa y así lo considera el ordenador que pasa a la línea siguiente del programa.

AND como una función

Como función, AND puede operar relacionando dos valores numéricos, por ejemplo:

50 **LET** x=y **AND** z

AND retorna el primer valor (y) si el segundo valor (z) no es igual a 0, y retorna 0 si el segundo valor (z) es 0. AND también puede operar con un valor de cadena (string value) siempre y cuando éste preceda a dicha palabra clave. Los valores numéricos deben ir siempre detrás de AND, por ejemplo:

50 **LET** a$=b$ **AND** z

AND retorna el primer valor (b$) si el segundo valor (z) no es cero. AND retornará una cadena nula o vacía si el segundo valor (z) es cero.

Observe que el INVES SPECTRUM+ asigna un valor de 1 a una condición verdadera y un valor de 0 a una condición falsa. El ordenador reconoce cualquier valor que no sea cero como verdadero y el valor cero como falso. El INVES SPECTRUM+ no evalúa combinaciones de valores numéricos de acuerdo con las tablas de veracidad estándar.

AND retorna el primer valor (b$) si el segundo valor (z) no es cero.

AND retornará una cadena nula o vacía si el segundo valor (z) es cero.

Observe que el INVES SPECTRUM+ asigna un valor de 1 a una condición verdadera y un valor de 0 a una condición falsa. El ordenador reconoce cualquier valor que no sea cero como verdadero y el valor cero como falso. El INVES SPECTRUM+ no evalúa combinaciones de valores numéricos de acuerdo con las tablas de veracidad estándar.

Formato

cond. **AND** cond.
expr-num **AND** expr-num.
expr de cadena **ANO** expr-num.

ASN Arc Sine — Arco Seno

Posición en el teclado

MODO EXTENDIDO
CAMBIO Q

Función

ASN calcula el valor de un ángulo a partir de su seno.

Cómo emplear ASN

ASN debe ir seguido por un valor numérico. Si es una expresión, ésta debe ir entre paréntesis. Por ejemplo:

60 **LET** x=**ASN**(y*z)

El valor que sigue a la palabra clave ASN (en el ejemplo y*z) es el seno del ángulo deseado, y puede variar entre +1 y —1. Obedeciendo la palabra clave ASN, el ordenador retornará el valor del ángulo en radianes (no en grados). Para convertir los radianes en grados se multiplica el valor devuelto por ASN por 180/PI.

Ejemplo

PRINT 180/PI* ASN 0.5

el resultado será 30, el ángulo en grados equivalente a un seno de 0.5.

Formato

ASN const-num
ASN var-num
ASN (expr-num)

AT

Posición en el teclado

CAMBIO I

Ver: **INPUT; LPRINT; PRINT**

ATN Arc TaNgent — Arco Tangente

Posición en el teclado

MODO EXTENDIO
CAMBIO E

Función

ATN le indica al ordenador que calcule un ángulo a partir de su tangente.

Cómo emplear ATN

La palabra clave ATN debe ir seguida por un valor numérico. La expresión debe ir ENCERRADA ENTRE PARENTESIS. Por ejemplo:

60 LET x=ATN(y*z)

El valor a continuación de ATN (en el ejemplo: y*z) es la tangente del ángulo requerido. ATN retornará el valor del ángulo en radianes (no en grados). Para convertir los radianes a grados multiplique el resultado suministrado por ATN por 180/PI.

Ejemplo

PRINT 180/PI * **ATN** 1

retornará el resultado 45, que es el ángulo en grados correspondiente a una tangente de 1.

Formato`

ATN const-num
ATN var-num
ATN (expr-num)

ATTR ATTRibutes — Atributos

Posición en el teclado

MODO EXTENDIDO
CAMBIO L

Función

ATTR suministra información sobre los atributos de una determinada posición de carácter en la pantalla. Los atributos son los colores de la tinta y el papel, y si los efectos de brillo, e intermitencia se encuentran operando o no.

Cómo emplear ATTR

ATTR debe ir seguido de dos valores numéricos. SEPARADOS POR UNA COMA Y ENCERRADOS ENTRE PARENTESIS. Por ejemplo:

150 **IF** ATTR(v,h) = 115 **THEN GOSUB** 2000

El primer valor a continuación de la palabra clave ATTR ("v» vertical) puede variar entre. *0* y 23, y se refiere a la posición de una de las líneas de la pantalla de televisión. El segundo valor («h» horizontal) puede variar entre *0* y 31 y es el número de la columna correspondiente. ATTR devuelve un número que es la suma de los atributos existentes en el punto

especificado. Este número se compone de los siguientes elementos:

Color de la tinta:	código de color (entre 0 y 7)
Color del papel:	8 veces el código de color.
Brillo:	64.
Intermitencia (Flash)	128

Ejemplo

Supongamos que un carácter situado en la posición 11,16 es visualizado en un color de tinta número 3 (magenta), color de papel número 6 (amarillo), y es brillante pero no intermitente. En este caso cuando el ordenador recibe el comando

PRINT ATTR (11,16)

Responderá: 115(3 + 8 x 6 + 64 + 0)

ATTR en forma binaria

ATTR retorna un byte. Si el bit 7 de este byte (muy significativo) es 1 esto indica que el efecto de intermitencia se encuentra en operación y 0 que está apagado. Lo mismo sucede en el caso del bit 6 que se refiere al efecto de brillo: el bit 1 indica que se encuentra en operación, y el bit 0 que está apagado. Los bits 5 a 3 se refieren al color del papel (en binario), y los bits de 2 a 0 ofrecen información sobre la tinta.

Formato
ATTR(expr-num, expr-num)

BEEP

Posición en el teclado

MODO EXTENDIDO
CAMBIO Z

Instrucción/Comando

La palabra clave BEEP causa que el altavoz del ordenador emita una nota de duración y tono predeterminados.

Cómo emplear BEEP

BEEP puede utilizarse para formar una instrucción incluida en un programa o también se puede emplear como comando directo. Esta palabra clave debe ir seguida por dos valores numéricos SEPARADOS POR UNA COMA, por ejemplo:

80 **BEEP** x,y

El primer valor (x) puede variar entre 0 y 10. Su propósito es definir la duración en segundos de la nota emitida. El segundo valor (y) puede variar entre -60 a + 69 y define el tono de la nota, en semitonos. Los valores negativos son por debajo del «do» central, y los positivos por encima del «do» central.

Ejemplo

El comando

BEEP 0 5, 1

producirá la nota "do" sostenido por encima del do central, con una duración de medio segundo.

Formato

BEEP expr-num, expr-num

BIN BINary number — Número BINario

Posición en el teclado

MODO EXTENDIDO
B

BIN transforma un número binario en un número decimal.

Cómo emplear BIN

BIN debe ir seguido por un número binario. Este número binario puede constar de hasta dieciséis números, 1 ó 0, por ejemplo:

50 **POKE USR** «a», **BIN** 10101010

La palabra clave producirá el número decimal correspondiente al número binario. BIN se utiliza comúnmente en conjunción de las palabras clave POKE y USR para crear caracteres gráficos definidos por el usuario del ordenador, como se hacía en el ejemplo anterior. En este caso, 1 significa un pixel de color de tinta y un 0 significa un pixel de color de papel (fondo).

Ejemplo

El comando

PRINT BIN 11111110

visualizará el número decimal 256 que es el equivalente del valor binario indicado en la instrucción.

Formato

BIN (1) [0]

BORDER — Borde

Posición en el teclado

B.

Instrucción/Comando

BORDER especifica el color del borde en torno del área visualización de la pantalla.

Cómo emplear BORDER

La palabra clave BORDER puede utilizarse como comando directo o bien como instruí-

ción incluida en un programa.BORDER debe ir seguido por un valor numérico, por ejemplo:

```
30 BORDER RND*7
```

El valor que aparece a continuación de la palabra clave BORDER se redondea al número entero más próximo. Su función es especificar el color del borde como sigue:

0 Negro
1 Azul
2 Rojo
3 Magenta (morado)
4 Verde
5 Azul-claro
6 Amarillo
7 Blanco

Observe que BORDER también define el color del papel (fondo) en la parte inferior de la pantalla. A diferencia de lo que ocurre en el caso de las palabras clave INK y PAPER, las instrucciones que contienen la palabra clave BORDER no pueden insertarse en instrucciones que contengan la palabra clave PRINT.

Ejemplo

BORDER 2

produce un borde de color rojo en torno del área de visualización de la pantalla.

Formato

BORDER expr número entero.

BRIGHT BRILLO — Brillo

Posición en el teclado

MODO EXTENDIDO
CAMBIO B

Instrucción/Comando

BRIGHT incrementa el brillo de los colores de los caracteres representados por el ordenador en la pantalla.

Cómo emplear BRIGHT

La palabra clave BRIGHT puede utilizarse como comando directo, pero normalmente se emplea para formar una instrucción de un programa. BRIGHT debe ir seguido de un valor numérico. Por ejemplo:

```
80 BRIGHT 1
```

El valor que está a continuación de BRIGHT se redondea al número entero más próximo, entre 1 y 8, si es necesario. Un valor de 1 determina que todos los caracteres visualizados en la pantalla a partir de ese momento en el que se ejecutan las instrucciones PRINT o INPUT aparezcan con los colores de la tinta (caracteres) y del papel (fondo) más brillantes. El valor 8 hace que al visualizarse nuevos caracteres en las posiciones de los caracteres que eran brillantes se sigan manteniendo brillantes, y los que se colocan en posiciones cuyos caracteres eran normales, se mantengan normales. Si la palabra clave BRIGHT va seguida de 0, se cancelarán las instrucciones BRIGHT 1 y BRIGHT 8, y el resultado es que todos los caracteres visualizados a continuación tendrán un brillo normal.

La palabra clave BRIGHT también puede insertarse dentro de instrucciones que afecten la visualización de la información en la pantalla, y que incluyan palabras clave como PRINT, INPUT, PLOT, DRAW, CIRCLE. La palabra clave BRIGHT se coloca a continuación de una de las palabras clave anteriores, pero debe preceder a los parámetros de datos o de visualización. BRIGHT va seguido por dichos valores y un punto y coma, por ejemplo:

```
50 PRINT BRIGHT 1; «ADVERTENCIA»
```

El efecto de BRIGHT será entonces local y se aplicará exclusivamente a los caracteres visualizados, como puntos o líneas trazados por el ordenador en la pantalla, mientras se ejecutaba ese comando. Recuerde que BRIGHT 1 aumenta el brillo del color del papel (fondo) de la totalidad de la posición del carácter, que es de 8 x 8 pixels, si cualquiera de los pixels de esa posición se dibuja en color tinta (caracteres).

Formato

BRIGHT Expr número entero [;]

CAT CATalogue — Catálogo

Comando para el manejo de la información almacenada en Microdrives. Véase el Manual sobre Microdrives e Interface 1.

CHR$ ChaRacterstring — Cadena de caracteres

Posición en el teclado

MODO EXTENDIDO
U

Función

El conjunto de caracteres del INVES SPECTRUM+ está formado por los caracteres y palabras clave que se encuentran en el teclado del ordenador, más cualquier carácter gráfico definido por el usuario. Se puede obtener cada uno de los componentes de este conjunto como una cadena mediante el empleo de la palabra clave CHR$ y un número de código. Este conjunto de caracteres también incluye varios códigos de control que afectan la visualización de los caracteres en la pantalla. Estos códigos se pueden poner en operación y se pueden visualizar los caracteres en la pantalla incorporando la palabra clave PRINT antes de CHR$. Al principio de este capítulo

se detalla el conjunto de caracteres completo y sus números de código.

Como emplear CHR$

CHR$ debe ir seguido por un valor numérico, par ejemplo:

80 **PRINT CHR$** x

Si es una expresión, esta debe ir encerrada entre paréntesis. El valor que va a continuaclan de CHR$ (en el ejemplo, x) se redondea el numero entero mas próximo. Los números que van de 32 a 255 producen en forma de cadena algunos de los caracteres del teclado del ordenador, elementos gráficos definidos par el usuario, o una palabra cave. Para valores entre 32 y 95 y entre 97 y 126, el INVES SPECTRUM+ emplea valores en código ASCII. En el ejemplo, si se asigna un valor de 65 a la variable x, el ordenador visualizara la letra A.

Códigos de control CHR$

Los valores entre 1 y 31 solo se utilizan para producir códigos de control. Las instrucciones CHR$6 (PRINT coma), 8 (retroceder un espacio) y 13 (nueva línea o ENTER) afectan a las imágenes visualizadas en la pantalla si están incluidas en una instrucción que lleve la palabra cave PRINT. CHR$ puede ir seguido por un número de código y un punto y coma, por ejemplo.

60 **PRINT** «A»; **CHR$**6; «B»

Esta instrucción le indica al ordenador que visualice lo siguiente:

A B

Los códigos de control CHR$ también pueden utilizarse para formar una cadena compuesta que los contenga. La instrucción

60 **PRINT** «A» + **CHR$**6 + «B»

tiene exactamente el mismo efecto que la anterior.

Los códigos que van desde el 16 al 23 afectan al color y la posición, y se pueden incluir en cadenas compuestas conjuntamente con CHR$. En este último caso, CHR$ 16 (control de la tinta o INK) y CHR$ 17 (control del papel o PAPER) irán seguidos de un número de código para los colores, entre 0 y 7; y en el caso de las palabras clave para los efectos especiales (FLASH, BRIGHT, INVERSE y OVER) los CHR$ 18 a CHR$21 irán seguidos por el número 0 (quo pone en funcionamiento el efecto especial) o 1 (quo lo termina). Por ejemplo, el comando:

PRINT CHR$**16+**CHR$3+**CHR$**17
+**CHR$**6+**CHR$**18+**CHR$**1
+«INVES SPECTRUM+»

tendrá como resultado la visualización de las palabras INVES SPECTRUM + en letras rojas y amarillas y con destellos. Como en los ejemplos anteriores, los símbolos + se pueden sustituir par puntos y comas.

CHR$22 (control AT) ira seguido de dos valores CHR$ para indicar los numeros de la línea y la columna. El comando:

PRINT CHR$**22+**CHR$11+**CHR$**16+**CHR$**42

le indica al ordenador que visualice una estrella en el centro de la pantalla.

CHR$23 (control TAB) va también seguido del mismo modo por dos valores. El segundo valor es normalmente 0 y el primero indica la posición de la tabulación, TAB.

PRINT CHR$**23+**CHR$16+**CHR$**0+**CHR$**42

produce la estrella en otra posición.

Tenga en cuenta quo estos valores son los únicos controles disponibles. Si se utiliza PRINT CHR$ con un valor de palabra clave mayor de 164, solo se visualizará el nombre de la palabra clave, sin que ésta se ejecute.

Formato

CHR$ const número entero [;] [+]
CHR$ var número entero [;] [+]
CHR$ (expr número entero) [;] [+]

CIRCLE — Círculo

Posición en el teclado

MODO EXTENDIDO
CAMBIO H

Instrucción/Comando

CIRCLE dibuja un círculo en la pantalla.

Cómo emplear CIRCLE

La palabra CIRCLE va seguida par tres valores numéricos, cada uno de ellos SEPARADO FOR UNA COMA. Ejemplo:

80 **CIRCLE**x,y,z

Si fuera necesario se redondearía cada uno de los tres valores al número entero más cercano. El ordenador traza un circulo en la cuadrícula de gráficos de alta resolución utilizando el color de tinta vigente en ese momento. Los dos primeros valores (x,y) definen las coordenadas horizontal y vertical del centro del circulo. El tercer valor (z) define la longitud del radio. Las dimensiones deberán ser tales que el círculo no se extienda fuera del área de visualización de la pantalla.

CIRCLE se ve afectado por instrucciones o comandos que controlan el color de las figuras de la pantalla, y puede incluir instrucciones referentes al color, con los mismos efectos que en las instrucciones que contienen las palabras clave PLOT y DRAW.

CIRCLE 128,88,87

dibujará un círculo ocupando la mayor parte del área de visualización.

Formato

CIRCLE (Instrucción;) número entero, número entero, número entero

CLEAR Borrar

Posición en el teclado
X

Instrucción/Comando

CLEAR borra los valores vigentes en ese momento, para todas las variables, liberando el espacio que éstas ocupan en la memoria. También libera el espacio de memoria hasta el valor RAMTOP, dirección superior del área controlada por el sistema BASIC. La palabra clave CLEAR también se puede utilizar para reajustar el valor RAMTOP (literalmente RAMTOP significa «el final del área RAM»).

Cómo emplear CLEAR

CLEAR puede utilizarse como comando directo, o bien incluirse dentro de una instrucción de un programa. No necesita parámetros, por ejemplo:

50 **CLEAR**

Como respuesta a la instrucción CLEAR, el ordenador borra los valores asignados en ese momento a todas las variables, incluidas las matrices (arrays).

También ejecuta las palabras clave CLS y RESTORE para limpiar la pantalla y restaurar el indicador de información o de datos (data pointer) para que señale al primer elemento de datos o información. Además, reajusta la posición de PLOT (dibujo), colocándola en el punto inferior izquierdo del área de visualización. También borra la pila GOSUB.

Observe que no se requiere incluir CLEAR antes de proceder a dimensionar matrices (arrays), ya que la palabra clave DIM utilizada en ese caso borra cualquier otra matriz preexistente de su mismo nombre.

Tenga presente, además, que RUN ejecuta la instrucción o comando CLEAR.

CLEAR y RAMTOP

CLEAR puede ir seguida por un valor numérico, por ejemplo:

CLEAR 65267

Con ese valor, CLEAR opera como se explicó anteriormente, y a continuación ajusta el RAMTOP —la dirección más alta del área del BASIC— al valor indicado. En el INVES SPECTRUM+, el valor máximo de RAMTOP se ajusta en 65367, y se encuentra ubicado debajo del área reservada para los gráficos definidos por el usuario. La palabra clave NEW borra el contenido de la memoria hasta el límite establecido en el RAMTOP. Como resoltado, podremos recurrir al empleo de la palabra clave CLEAR para definir un sector de la memoria inmune a la palabra clave NEW. En el ejemplo, esta área se encuentra entre el lí-mite máximo (RAMTOP) del ordenador INVES SPECTRUM+ (65267) y el límite establecido mediante la palabra clave CLEAR (65267) y su capacidad es de 100 bytes. Si elevamos el límite RAMTOP se incrementará la memoria disponible para el BASIC, sacrificando parte del espacio reservado para los gráficos definidos por el usuario. Observe que la pila GOSUB se situará en el RAMTOP siempre y cuando el RAMTOP permanezca por encima de la pila del calculador.

Para determinar cual es el RAMTOP, introduzca en el ordenador las instrucciones siguientes.

PRINT PEEK 23730+256***PEEK**23731

Formato

CLEAR (exp-num)

CLOSE

Comando para el manejo de información almacenada en Microdrives. Ver el manual sobre Microdrives e Interface 1.

CLS Clear Screen Limpiar Pantalla

Posición en el teclado

V

Instrucción/Comando

CLS borra todos los textos y gráficos en el área de visualización de la pantalla, dejándola en blanco y con el color de papel (fondo) vigente.

Cómo emplear CLS

CLS puede emplearse como comando directo o bien como elemento de una instrucción incluida en un programa. No requiere parámetros. Por ejemplo:

250 **IF** a$= «NO» **THEN CLS**

Si se cumple la condición estipulada, el or denador borrará el área de visualización (pero no el borde de la pantalla), adoptando el color del papel seleccionado por el comando o instrucción PAPER inmediatamente anterior. Si el programa no incluye esta instrucción, el ordenador adoptará como color de papel PAPER el color blanco.

Observe que CLS debe utilizarse después de la palabra clave PAPER y antes de PRINT, o cualquier otra instrucción que afecte a la visualización, si lo que se desea es producir un fondo de colores que abarque toda el área de visualización.

Formato

CLS

CODE — Código

Posición en el teclado

MODO EXTENDIDO
I

Función

La función CODE suministra el número de código de un carácter determinado, dentro del conjunto de caracteres del INVES SPECTRUM+ (ver la tabla al principio de este capítulo).

Cómo emplear CODE

La función CODE debe ir seguida por una cadena (string), por ejemplo:

90 **IF CODE** a$<65 **OR CODE** a$>90 **THEN** GOTO 80

Las expresiones que forman parte de la cadena deben ir ENCERRADAS ENTRE PARENTESIS. CODE suministra el número de código del primer carácter de la cadena. Si la cadena es una «cadena vacía», CODE suministra el valor 0.

CODE suministra los números de código ASCII, desde el 32 al 95 y desde el 97 al 126.

Ejemplo

PRINT CODE "INVES SPECTRUM+"

visualizará en la pantalla el código 73, correspondiente a la letra I.

SAVE CODE y LOAD CODE

CODE se utiliza de forma diferente cuando se combina con SAVE y LOAD. (Para grabación y carga de rutinas o programas en código máquina).

Formato

CODE const de cadena
CODE var de cadena
CODE (exp de cadena)

CONTINUE — Continuar

Posición en el teclado

C

Comando

Si el programa se detiene, la palabra clave CONTINUE puede ser utilizada para volverlo a iniciar a partir del punto donde se detuvo. Si la detención en la ejecución del programa se debe a un error, éste deberá ser rectificado antes de que CONTINUE permita que el programa se reinicie.

Cómo emplear CONTINUE

Esta palabra clave se utiliza como comando directo cuando se ha detenido un programa. No necesita parámetros. Normalmente, después de CONTINUE, el programa se reinicia en la instrucción donde se produjo la interrupción. Si la causa de ésta fue un error, se podrá introducir un comando para rectificarlo y CONTINUE permitirá que el programa continúe a partir de esa instrucción. Si el programa se ha detenido en una instrucción que contenga la palabra clave STOP, causando con ello un mensaje 9, o si se detuvo debido a que se oprimió la tecla PARAR, causando un mensaje L, CONTINUE reiniciará el programa a partir de la instrucción siguiente. Antes de introducir CONTINUE, y si es encesario, se puede teclear otro comando para rectificar la situación.

Cuando se emplea la palabra clave CONTINUE para reasumir un comando directo, esta palabra clave entrará en un bucle (loop) si el comando se detuvo en la primera instrucción del comando. En este caso, la imagen desaparece pero se puede recuperar el control pulsando la tecla PARAR. CONTINUE causa un mensaje 0 si el comando se detuvo en la segunda instrucción, y un mensaje N si se detuvo en la tercera instrucción, o siguientes.

Formato

CONTINUE

COPY — Copiar

Posición en el teclado

Z

Comando

COPY hace que una impresora que disponga de HARDCOPY reproduzca una copia exacta del contenido actual de la pantalla.

Cómo emplear COPY

La palabra clave COPY se utiliza como comando directo cuando se ha ejecutado completamente un programa o cuando se ha detenido. No requiere parámetros. Si la impresora se encuentra conectada y funcionando, después del comando COPY, se producirá la copia de las primeras 22 líneas de la imagen en la pantalla. Los colores de tinta (caracteres de escritura) se imprimirán en negro, los colores de papel (fondo) no se imprimen. La impresora puede detenerse oprimiendo la tecla PARAR.

Una vez visualizado el listado de un programa en la pantalla, puede copiarse mediánte la utilización del comando COPY, si el listado se produjo mediante un comando o instrucción LIST. Recuerde que después de completar o detener un programa podemos hacer que aparezca el listado en la pantalla

presionando la tecla ENTER. Sin embargo, en este último caso de «listado automático», el listado no podrá copiarse mediante el comando COPY.

Formato

COPY

COS COSine Coseno

Posición en el teclado

MODO EXTENDIDO
W

Función

COS suministra el coseno de un ángulo

Cómo emplear COS

COS debe ir seguido por un valor numérico, por ejemplo:

140 **LET**x=**COS**y

Si es una expresión, ésta debe ir entre paréntesis. El valor que se encuentra a continuación de COS es el ángulo expresado en radianes. COS suministra el coseno del ángulo. Los grados pueden convertirse en radianes multiplicándolos por P1/1 80.

Observe que COS suministra un valor negativo para ángulos de entre 90 a 270 grados, y un valor positivo para ángulos entre 0-90 y 270-360 grados.

Ejemplo

El comando

PRINT COS (60*__PI__/180)

produce como resultado el valor 0.5, el co-seno correspondiente a un ángulo de 60 grados.

Formato

COS const-num
COS var-num
COS (expr-num)

DATA Datos

Posición en el teclado

MODO EXTENDIDO
D

Instrucción

La instrucción DATA suministra una lista de elementos de información, o datos dentro de un programa. Estos elementos pueden ser, por ejemplo los valores adjudicados a las variables o a cadenas que visualizará el ordenador. Cada elemento de información se asigna a una variable mediante una instrucción READ.

La asignación de datos o información a las variables del programa se efectúa siguiendo el orden en el que aparecen dichos datos en el programa. Sin embargo, se puede utilizar la palabra clave RESTORE para comenzar la asignación de datos a variables partiendo del primer elemento de información contenido en una instrucción DATA.

Cómo emplear DATA

DATA sólo puede emplearse para formar una instrucción que forme parte de un programa. Normalmente, esta palabra clave va seguida por una lista de constantes numéricas o de cadena SEPARADAS ENTRE ELLAS POR COMAS. Por ejemplo:

50 **DATA** 31, «ENER0»,28,«FEBRERO»

Mediante una instrucción READ, se asigna cada una de las constantes a una variable determinada, indicándole de este modo al ordenador que debe leer un elemento de información o dato. Estas instrucciones DATA pueden colocarse en cualquier parte del programa. El número, tipo (numérica o de cadena) y orden de las constantes debe corresponder al número de veces que el ordenador debe ejec.-tar la instrucción READ. La naturaleza y orden de las variables dentro de la instrucción READ también debe estar de acuerdo con los elementos de la instrucción DATA. Si existen demasiados elementos de información para una misma instrucción, la lista de datos puede desdoblarse en varias instrucciones DATA sucesivas.

Ejemplo

El programa siguiente:

```
10 FOR n=1 TO 2
20 READ x,a$
30 PRINT a$,x; «días»
40 NEXT n
50 DATA 31, «Enero», 28,«Febrero»
```

visualizará lo siguiente en la pantalla:

```
Enero        31 días
Febrero      28 días
```

Utilización de DATA con variables

Cada elemento de información o de datos incluido en una instrucción DATA puede consistir en variables numéricas, o de cadena, o bien expresiones, siempre y cuando se les haya asignado valores. En el ejemplo anterior, la instrucción DATA se puede modificar del siguiente modo:

50 **DATA** d,m$,d-3, «Febrero»

Si previamente se asignó el valor 31 a la variable d, y el valor «Enero» a la variable m$, al ejecutar la instrucción se obtendrá exactamente el mismo resultado que antes.

LOAD DATA, SAVE DATA y VERIFY DATA

La palabra clave DATA puede también utilizarse con LOAD, SAVE y VERIFY para almacenar matrices (arrays) en cintas de cassette.

Formato

DATA expr-num [,expr-num] [expr de cadena]
DATA expr de cadena [,expr-num] [,exp.-num]

DEF FN DEFinir
DEFine FuNction FuNcion

Posición en el teclado

MODO EXTENDIDO
CAMBIO 1

Instrucción

DEF FN habilita al usuario para definir una función que se encuentra disponible como tal palabra clave. En una instrucción FN se pueden pasar ciertos parámetros a la función. La instrucción FN reclama la función y puede producir como resultado un valor de cadena o numérico.

Cómo emplear DEF FN

DEF FN sólo puede utilizarse como instruíción contenida en un programa. Cuando se desee definir una función numérica, DEF FN debe ir seguida por cualquier letra individual y a continuación por una o más variables numéricas SEPARADAS UNAS DE OTRAS POR COMAS, y ENCERRADAS ENTRE PARENTESIS, por ejemplo: DEF FN r (x,y). A continuación se encuentra un signo igual y una expresión numérica que contenga las variables, por ejemplo:

1000 **DEF FN**r(x,y) =**SQR**(x ↑2+y ↓ 2)

La letra a continuación de DEF FN (r en el ejemplo) es un nombre que identifica la función. Las variables también pueden consistir en letras individuales. Observe que en ambos casos el INVES SPECTRUM+ no distingue entre letras mayúsculas y minúsculas.

La expresión que sigue al signo igual utiliza las variables (x,y) para definir la función.

Las instrucciones DEF FN pueden colocarse en cualquier parte dentro del programa.

Para llamar a la función definida por DEF FN, se utiliza la palabra clave FN seguida por la letra que designa la función y una lista de valores numéricos, cada uno de los cuales debe ir SEPARADO DE LOS DEMAS POR UNA COMA, y DEBE IR ADEMAS ENTRE PARENTESIS, por ejemplo:

50 **PRINT FN**r(3,4)

Los valores numéricos entre paréntesis son transferidos a la función en el mismo orden en que se encuentran las variables en la sentencia DEF FN.

Así, en el ejemplo anterior, x recibirá un valor de 3, mientras que y recibirá un valor de 4.

FN evalúa la expresión y produce como resultado un valor.

DEF FN también puede ir seguida por una letra y un par de paréntesis únicamente, por ejemplo:

1000 **DEF FN**r( )=**INT**(x+0 5)

El valor asignado en este momento a la variable (x en el ejemplo) es pasado a la función al ser llamado por la palabra clave FN.

En este caso, FNr( ) produce el valor asignado a x en este momento, redondeándolo al número entero más próximo.

DEF FN y cadenas (string)

DEF FN también se puede utilizar del mismo modo, con la intención de definir e invocar una función de cadena (string function). En este caso, el nombre de la función es una sola letra seguida por el signo de dólar $.

La definición está formada por una expresión de cadena (string expression).

1000 **DEF FN** aS(b$,x,y)=b$(x TOY)

En este ejemplo, la expresión de cadena que está a continuación del signo igual utiliza un fragmentador de cadena, y las variables x e y son el primero y último carácter de un pedazo de b$.

FN debe ir seguida por el nombre de la función y un valor de cadena entre paréntesis, conjuntamente con cualquier otro parámetro que debe pasarse a la función.

En este caso, el comando.

PRINT FN A$ (FUNDAMENTAL» 1,3)

Produce como resultado que el ordenador visualice en la pantalla: FUN. Del mismo modo, el comando:

PRINT FN a$ (“FUNDAMENTAL»,5,8)

producirá en la pantalla: AMEN.

Formato

DEF FN letra ([letral].[, letra])=expr-num
DEF FN letra$ ([letra$] [letra] [,letra] [,letras$]=expr de cadena
FN letra ([expr-num] [,expr-num])

FN letra$ ([expr de cadena] [expr-num] [,expr-num] [,expr de cadena])

DIM DIMension Dimensión

Posición en el teclado

D

Instrucción

La palabra clave DIM se utiliza para establecer las dimensiones (para crear) una matriz (array) que contenga una cantidad predeterminada de variables numéricas o de cadena.

Una matriz contiene un juego de variables, todas con el mismo nombre, y que se distinguen las unas de las otras por sus subíndices (en inglés: subscripts). Los subíndices son valores que identifican cada una de las variables o elementos en la matriz.

Cómo emplear DIM con matrices numéricas

DIM se utiliza para formar una instrucción, dentro de un programa. Debe ir seguida por una letra individual que designa la matriz, y uno o más valores numéricos. Cada valor numérico debe ir SEPARADO POR UNA COMA Y ENCERRADO ENTRE PARENTESIS. Por ejemplo:

```
20 DIMx(10)
80 DIMz(20,5)
```

La primera es una matriz unidimensional que contiene diez elementos con subíndices que van del 1 al 10. El nombre de la matriz es x y sus variables se identifican combinando el nombre de la matriz con los respectivos subíndices, por ejemplo: x(1), x(3), x(10), etc.

Cuando el ordenador llega a esta instrucción en el programa, borra cualquier matriz que tenga en su memoria que lleve el mismo nombre y asigna un valor 0 a cada una de las variables de la matriz.

Tenga presente que al establecer las dimensiones de una matriz el INVES SPECTRUM+ no distingue entre letras mayúsculas y minúsculas —para el ordenador la variable x(2) es la misma que X(2). Sin embargo, las variables numéricas simples, cuya letra sea igual al nombre de una matriz (en nuestro ejemplo x o X) pueden coexistir con esta última, y si es necesario se pueden utilizar separadamente.

El número de valores entre paréntesis debe ser igual a las dimensiones que se han dado a la matriz numérica.

El segundo ejemplo muestra una matriz bidimensional con un total de 100 elementos: 20 elementos en la primera dimensión y 5 en la segunda (el total es 20 x 5). En este caso los elementos se numeran z(1,1) hasta z(20,5).

Pueden crearse matrices de cualquier número de dimensiones.

Los elementos de una matriz numérica pueden ser identificados subsecuentemente por el nombre de la matriz seguida por un valor en-tre paréntesis, por ejemplo:

```
 70 PRINT x(a)
160 PRINT z(7,b)
```

DIM y matrices de cadena

La palabra clave DIM es utilizada de la misma forma en las matrices numéricas, con la diferencia de que se utiliza una letra individual seguida por el signo $ para nombrar la matriz.

Además, debe agregarse un valor extra a los valores entre paréntesis para definir el largo de cada cadena. Por ejemplo:

```
30 DIM a$(20,5)
90 DIMb$(20,5,10)
```

La primera sentencia crea una matriz de 20 elementos, cada una de las cuales contiene una cadena de cinco caracteres. Las variables identificadas por subíndices se designan desde a$(1) hasta a$(20) inclusive. Al Princ.-pio, el ordenador asigna a cada variable una cadena «vacía» o «nula» (« »).

Como antes, el ordenador borrará de su memoria cualquier otra matriz del mismo nombre que pueda encontrarse en ella. Además, a diferencia de lo que sucede en el caso de las matrices numéricas, una variable de cadena simple no puede coexistir con una matriz de cadena del mismo nombre.

El segundo ejemplo crea una matriz de cadena bi-dimensional, con un total de 100 elementos, 20 elementos en la primera dimensión y 5 en la segunda. Todos los elementos tienen una extensión de 10 caracteres.

Del mismo modo, cuando se asignan valores a una matriz de cadena, éstos se completentan con otros valores de relleno al final de la cadena o bien se cortan si es necesario para ajustarlos a la extensión especificada.

Los elementos de una matriz de cadena son identificados por el nombre de la matriz, seguido por uno o más valores numéricos entre paréntesis, proporcionando el número o números del subíndice correspondiente.

Por ejemplo, el elemento a$(2) de la matriz puede ser «PEREZ», y el elemento b$(12,4) puede ser «MADRID».

Sin embargo, es posible agregar un valor extra para definir un carácter particular en la cadena. En estos ejemplos a$(12,4,5) sería R (el cuarto carácter en MADRID).

Matrices de cadena de dimensión cero

Se pueden crear matrices de cadena de dimensión cero utilizando únicamente un valor entre paréntesis, por ejemplo:

```
10 DIMc$(15)
```

Esta matriz contiene únicamente un elemento c$, y su extensión se establece en el valor definido de 15 caracteres (15).

Formato

DIM letra (expr-núm [,expr-núm])
DIM letra$ (expr-núm [,expr-núm])

DRAW — Trazar

Posición en el teclado

W

Instrucción/Comando

La palabra clave DRAW se utiliza para trazar líneas rectas o curvas en la pantalla.

Cómo emplear DRAW

Normalmente, DRAW se utiliza para formar una instrucción, dentro de un programa. Si lo que se desea es trazar una línea recta, debe ir seguida de dos valores numéricos SEPARADOS POR UNA COMA, por ejemplo:

40 **DRAW**x,y

El ordenador trazará una línea recta en la cuadrícula de gráficos de alta resolución. La línea se trazará partiendo de la posición definida por la instrucción PLOT anterior, o la posición donde terminó la línea trazada por una instrucción DRAW anterior (la última).

Los dos valores a continuación de la palabra clave DRAW se redondearán al número entero más próximo si fuera necesario.

El primer valor (en el ejemplo, x) define la distancia en el sentido horizontal desde esta posición, y el segundo valor (y) define la distancia vertical. Estos valores serán negativos si la línea se traza hacia la izquierda o hacia abajo. La posición del final de la línea debe encontrarse dentro del área de visualización de la pantalla.

Cuando el programa no contiene sentencias de PLOT o DRAW anteriores, la nueva sentencia DRAW tendrá el efecto de comenzar a trazar la línea a partir de la esquina inferior izquierda de la pantalla, identificada por las coordenadas 0,0.

La sentencia DRAW se ve afectada por instrucciones que contienen comandos o instruíciones relativas al color. También puede llevar otras instrucciones incorporadas, con los mismos resultados que en las instrucciones que incluyen las palabras clave PLOT y CIRCLE.

DRAW, trazando curvas

DRAW puede ir seguida por un tercer valor y producirá una curva, o un sector de circunferencia, por ejemplo:

40 **DRAW**x,y,z

El tercer valor (z) define el ángulo en radianes de la línea trazada en la pantalla. Cuando el valor es positivo la línea será trazada hacia la izquierda, y hacia la derecha cuando es negativo. Los valores de PI o —PI producen un círculo.

Ejemplo

El programa siguiente traza un triángulo en la pantalla

10 **PLOT** 127, 150
20 **DRAW** 70,-100
30 **DRAW** -140,0
40 **DRAW** 70,100

Cuando se agrega un valor 1 o —1 a la instrucción DRAW, los lados se transforman en curvas hacia dentro o hacia fuera.

Formato

DRAW (instrucción;) expr número entero, expr número entero (,expr número entero)

ERASE Borrar

Comando para la operación de almacenamiento de información utilizando Microdrives. Vea el Manual de Microdrives e Interface 1.

EXP EXPonent Exponente

Posición en el teclado

X

Función

EXP es una función aritmética que eleva el exponente a una potencia determinada.

Cómo emplear EXP

La palabra clave EXP debe ir seguida por un valor numérico, por ejemplo:

60 **LET** y=**EXP**x

La expresión debe ir encerrada ENTRE PARENTESIS. Como resultado, EXP da al exponente elevado a la potencia del argumento (x en nuestro ejemplo).

Ejemplo

PRINT EXP1

produce el valor de e, que es:
2.7182818

Formato

EXP const-num
EXP var-num
EXP (expr-num)

FLASH Parpadeo

Posición en el teclado

MODO EXTENDIDO
CAMBIO V

Instrucción/Comando

La palabra clave FLASH le indica al ordenador que determinadas posiciones de caracteres "parpadeen", es decir, produzcan destellos. Este efecto óptico se obtiene alternando los colores de tinta (caracteres) y papel (fondo) a una velocidad predeterminada.

Cómo emplear FLASH

FLASH puede emplearse como comando directo, pero por lo general forma parte de una

instrucción dentro de un programa. Debe ir seguido de un valor numérico, por ejemplo:

```
50 FLASH 1
```

El valor numérico que está a continuación de FLASH se redondea al número entero más próximo (si es necesario), y puede ser 0, 1 u 8.

el valor 1 hace que parpadeen todos los caracteres que se visualicen a partir del momento de la ejecución de instrucciones que contengan las palabras clave PRINT e INPUT.

FLASH debe colocarse a continuación de la palabra clave, pero preceden a los parámetros de la información (datos) o de la imagen que se va a visualizar. FLASH va seguido por los mismos valores y un punto y coma.

Por ejemplo

```
120 PRINT FLASH 1; INK2; PAPER 6;
                              «ATENCION»
```

este caso el efecto de FLASH será local y se aplicará únicamente a los caracteres visualizados, punto o línea trazada en ejecución de esa sentencia.

Recuerde que FLASH determina el estado de la totalidad de la posición de 8 x 8 pixels, aunque solamente se dibuje (PLOT) en un color de tinta la posición de uno de ellos.

Formato

FLASH expr número entero [;]

FN FUNction — Función

Posición en el teclado

MODO EXTENDIDO
CAMBIO 2

Función

FN invoca una función definida por el usuario. Esta palabra clave siempre se utiliza en conjunción con DEF FN, que es la que define la función que va a ser invocada.

Cómo emplear FN

Si se trata de una función numérica, FN va seguido de una letra y un par de paréntesis. Cuando se desee pasar parámetros a la función, estos deben IR SEPARADOS POR COMAS Y ENCERRADOS ENTRE PARENTESIS. Por ejemplo:

```
170 LETx = FNr(3,4)
```

Los parámetros (3 y 4 en el ejemplo) se pasan a la función (denominada r en este caso). A continuación FN produce el resultado.

El par de paréntesis debe incluirse aún cuando no se requiera pasar parámetros a la función, por ejemplo:

```
70 PRINT FNr()
```

En este caso, la función empleará los valores asignados en ese momento a sus variables.

FN invoca una función de cadena de acuerdo con el mismo procedimiento, excepto que en este caso es necesario agregar el símbolo $ inmediatamente después de la letra que designa la función. Para más detalles consulte la sección sobre la palabra clave DEF FN.

Formato

FN letra ([expr-num] [,expr-num])
FN letra$ ([expr de cadena] [expr-num] [,expr-num] [,expr de cadena])

FOR — Para

Posición en el teclado

F

Instrucción/Comando

FOR se utiliza siempre con las otras dos palabras clave TO y NEXT para construir un bucle (loop) FOR NEXT. Esta estructura permite la repetición de una determinada sección del programa un número de veces predeterminado.

Cómo emplear FOR

FOR siempre forma una instrucción con TO. La palabra clave FOR va seguida de una letra, un signo igual y dos variables numéricas separadas por la palabra clave TO. Por ejemplo:

```
60 FOR a = 1 TO 9
```

La letra (a en el ejemplo) es una variable de control. A continuación se incluyen las instruíciones que deben repetirse; normalmente una o más instrucciones utilizan la variable de control. El bucle termina con la instrucción NEXT. En esta última instrucción, NEXT va seguido por el nombre de la variable de control (a en nuestro caso). Por ejemplo:

```
90 NEXT a
```

Al ejecutarse el programa, FOR cumple una serie de operaciones. Primero: borra cualquier variable del mismo nombre que la variable de control. Segundo, le asigna a ésta un valor inicial idéntico al valor especificado antes de TO (1 en el ejemplo). A continuación se ejec.-tan las instrucciones. Tercero: cuando el ordenador llega a la palabra clave NEXT incrementa el valor de la variable de control en una unidad (a 2 en nuestro ejemplo). Cuarto: el ordenador comparará el nuevo valor de la variable de control con el valor indicado después de TO (9). Este último valor establece el límite para el bucle. Quinto: si este valor es inferior al de límite, el ordenador reiniciará el proceso retornando a la instrucción FOR y repitiendo el bucle FOR NEXT. Al finalizar cada bucle el valor de la variable de control se incrementa en

una unidad. Cuando el ordenador comprueba que el valor de la variable de control es superior al límite (9), termina á bucle, A continuación el programa continuará con la instrucción que sigue a NEXT.

En el ejemplo el bucle se repite nueve veces, y el valor de la variable de control se incrementa desde 1 a 9. Al terminarse el último bucle, la variable de control tendrá el valor 10.

Recuerde que el INVES SPECTRUM+ no distingue entre letras mayúsculas y minúsculas cuando éstas se utilizan para designar una variable de control.

Utilización de STEP en un bucle FOR NEXT

La palabra STEP puede incorporarse a una instrucción para incrementar o disminuir el valor de la variable de control por pasos mayores de 1. STEP puede colocarse después del valor límite y va seguido por un valor numérico. Por ejemplo:

60 **FOR** a=1 **TO** 9 **STEP** 2

En este caso el valor de la variable de control será incrementado al cumplirse un bucle en la cantidad indicada por STEP (2 en el ejemplo). El bucle se repetirá hasta que el valor de la variable de control sea superior al valor límite establecido.

En el ejemplo los valores sucesivos de la variable de control serán 1,3,5,7, y 9. El ordenador abandonará el bucle cuando la variable de control alcance el valor de 11 (9+2).

STEP puede llevar un valor negativo, y con ello la variable de control disminuirá en lugar de incrementarse. Lógicamente, en este caso el valor inicial debe ser mayor que el valor límite. El bucle se terminará cuando el valor de la variable de control sea menor que el valor límite. Por ejemplo:

60 **FOR** a = 9 **TO** 1 **STEP**-1

El valor límite disminuye desde 9 a 1, y el ordenador abandona el bucle cuando el valor de la variable de control llega a 0.

Acomodando un bucle dentro de otro.

Esta operación se describe muy gráficamente como «anidar bucles». Se pueden anidar uno o más bucles FOR NEXT uno dentro de otro. En este caso, el orden de las variables de control de las instrucciones NEXT debe ser el inverso del orden de control de las instrucciones FOR. Se pueden anidar todos los bucles que se deseen.

Formato

FOR letra = expr-num **TO** expr-num
(**STEP** expr-num) **NEXT** letra

FORMAT — Formato

Este comando se utiliza para controlar el procedimiento de almacenamiento en los Microdrives. Consulte el Manual de Microdrives e Interface 1.

GOSUB — Ir a la subrutina

Posición en el teclado

H

Instrucción/Comando

La palabra clave GOSUB hace que el ordenador se desvíe a una subrutina. Las subrutinas son secciones de programa separadas normalmente de éste y compuestas de varias líneas. Este procedimiento es muy útil cuando la misma subrutina se utiliza varias veces durante le ejecución del programa.

Cómo empezar GOSUB

GOSUB puede emplearse como instrucción o comando directo. Va seguido por un valor numérico. Por ejemplo:

50 **GOSUB** 1000

Cuando el ordenador ejecuta el programa, el valor situado a continuación de GOSUB se redondea al número entero más próximo y el ordenador toma el desvío indicado dirigiéndose a la línea de programa que lleve ese número. El empleo de una variable o expresión permite que el ordenador tome el desvío o la subrutina, encaminándose a una linea predeterminada. Tenga presente que el ordenador se desviará al llegar a la sentencia GOSUB aún en el caso en el que el número de línea mencionado no se encuentre en el programa. En este caso, el ordenador continuará con la primera instrucción que encuentre en esa posición.

La subrutina se termina con la palabra clave RETURN (retorno). Al encontrarla, el ordenador retornará a la instrucción situada inmediatamente después de la instrucción que contiene la palabra case GOSUB que inició el proceso.

Las subrutinas pueden anidarse unas dentro de otras. En este caso, el ordenador se envía de una subrutina a otra En todos los casos, la palabra clave RETURN que termina la subrutina le indica al ordenador que debe retornar a la instrucción situada inmediatamente después de la última instrucción que contiene la palabra clave GOSUB (la ejecutada en último lugar).

La pila de GOSUB

Cuando se ejecuta una instrucción GOSUB, su número de línea se almacena en la pila GOSUB de la memoria del ordenador. Cuando se ejecutan dos o más sentencias GOSUB antes de encontrar un RETURN, el ordenador va almacenando los números de las líneas del programa donde van apareciendo las GOSUB

que va ejecutando. Estos números de línea se almacenan de tal modo que el último incluido se encuentre en la posición más alta de la pila. El efecto de cada palabra clave RETURN es que el ordenador vuelve al número de línea GOSUB que se encuentra en la parte más alta de la pila en cada momento. Como resultado de ello el ordenador volverá a la línea que sigue a la última instrucción GOSUB ejecutada.

Tenga presente que cuando no existen suficientes palabras clave RETURN puede producirse el error 4 (Out of memory, o «sin memoria».)

Formato

GOSUB expr número entero

GOTO — Ira la línea

Posición en el teclado

G

Instrucción/Comando

La palabras clave GOTO le indica al ordenador que se desvíe y se dirija a una línea determinada del programa (A diferencia de GOSUB que le indica que avance a una subrutina).

Cómo empezar GOTO

La palabra clave GOTO puede ser utilizada como comando directo para ejecutar un programa a partir de una línea determinada, SIN BORRAR PREVIAMENTE el contenido de la pantalla. También se puede emplear para formar una instrucción dentro de un programa. GOTO va seguida por un valor numérico, por ejemplo:

60 **GOTO** 350

Cuando se ejecuta la instrucción que contiene la palabra clave GOTO, el valor situado a continuación de GOTO se redondea al número entero más próximo, y el ordenador se desvía dirigiéndose a la línea indicada. El empleo de una variable o expresión permite que el programa se desvíe a una línea determinada. Tenga presente que el ordenador se desviará al llegar a la instrucción GOTO aun en el caso de que el número de línea mencionado en la instrucción no se encuentre en el programa. En este caso, el ordenador continúa con la primera instrucción que encuentre en esa posición.

Formato

GOTO expr número entero.

IF — Si

Posición en el teclado

U

Instrucción/Comando

La palabra clave IF siempre se utiliza conjuntamente con la palabra clave THEN (entonces) para producir una decisión que determina la acción siguiente. Antes de tomar la decisión, el ordenador examina ciertas condiciones para determinar si son verdaderas o falsas. El efecto de esas dos palabras clave es indicarle al ordenador que «si se cumplen ciertas condiciones, haga lo siguiente». Cuando llega a ese punto del programa, el ordenador se encuentra frente a una encrucijada y el programa le indica que si cierta condición es verdadera, debe tomar un camino, y si es falsa, debe tomar el otro.

Cómo emplear IF y THEN

Normalmente, IF forma parte de una instruíción con THEN. IF va seguido en primer lugar por un valor numérico o una condición, y en segundo lugar por THEN. A continuación va seguido por una o varias instrucciones válidas en lenguaje BASIC. Por ejemplo:

80 **IF** x **THEN GOTO** 250
240 **IF** a$=«NO» **THEN PRINT** «FIN»: **STOP**

Una constante, variable o expresión (por ejemplo, x) se considerará verdadera si tiene un valor diferente de 0. En este caso, la instrucción que está a continuación de THEN y las demás instrucciones de la misma línea del programa se ejecutarán. Después, el programa sigue a la línea que está a continuación de la instrucción que contiene la palabra clave THEN. Se considerarán falsas las constantes, variables o expresiones cuyo valor sea 0. En consecuencia, las instrucciones que están a continuación no se ejecutarán, y el programa continuará por la línea que contiene la palabra clave IF. En el ejemplo, el ordenador no continuará a la línea 250 si el valor de x es 0.

Si la condición especificada (a$=«NO») a continuación de la, palabra clave IF es verdadera, entonces el ordenador ejecutará las sentencias a continuación de THEN. Si la condición es falsa, el ordenador continuará a la próxima línea En el caso de nuestro ejemplo, si la variable a$ tiene el valor «NO», el ordenador visualizará el texto indicado entre comillas («FIN») y ejecutará la instrucción «STOP» poniendo fin al programa. En cambio, si a$ tiene cualquier otro valor, el ordenador continuará el programa saltando a la próxima línea.

El INVES SPECTRUM+ adjudica a una condición verdadera el valor 1 y a una condición falsa el valor 0. Como resultado, el ordenador reconoce cualquier valor, excepto el cero. como verdadero, y considera el 0 como falso. Puede asignarse el valor de una condición a una variable utilizando una instrucción como la siguiente:

70 **LET** x=a$=«NO»

Recuerde que, a diferencia de lo que sucede en otros dialectos del BASIC, en el INVES SPECTRUM+ la palabra clave THEN no puede ser omitida antes de GOTO.

Formato

IF expr-num - sentencia conteniendo **THEN** [sentencia]
IF condición **THEN** sentencia

IN — En

Posición en el teclado

MODO EXTENDIDO
CAMBIO I

Función

IN controla el estado del teclado y otros equipos que sirven para introducir información en el ordenador (input) o sacarla (output). Esta palabra clave lee un byte desde la dirección de una puerta (port address) determinada, indicando el estado del equipo o sistema conectado a esa puerta.

Cómo emplear IN

IN va seguido por un valor numérico, por ejemplo:

150 **LET** x=**IN** y

El valor que va después de IN puede variar entre 0 y 65535. La función de ese valor es indicar la dirección de la puerta que debe leer el ordenador. IN devuelve la lectura de esa puerta en bytes.

DIRECCIONES DEL TECLADO

El teclado tiene ocho direcciones. El contenido de cada una de ellas puede ser uno de entre cinco valores diferentes, dependiendo de la tecla que se oprima en cada momento. Las direcciones son las siguientes: 65278, 65022, 64510, 63486, 61438, 57342, 49510 y 32766. Los valores de los bytes en esas direcciones pueden ser: 175, 183, 187, 189 o 190.

Formato

IN num-const
IN num-var
IN (expr-num)

INK — Tinta

Posición en el teclado

MODO EXTENDIDO
CAMBIO X

Instrucción/Comando

La palabra clave INK especifica el color de los elementos que aparecen en primer plano en la pantalla. Estos elementos incluyen caracteres, puntos, líneas rectas y curvas.

Cómo emplear INK

INK puede emplearse como comando directo pero normalmente se utiliza como parte de una instrucción de programa. Debe ir seguido por un valor numérico, por ejemplo:

70 **INK** x

El valor numérico que se encuentra a continuación de INK se redondea al número entero más próximo y puede variar entre 0 y 9. Los códigos de los colores para los elementos del primer plano son los siguientes:

0	Negro
1	Azul
2	Rojo
3	Magenta (morado)
4	Verde
5	Azul-verde
6	Amarillo
7	Blanco
8	Transparente
9	Negro o blanco de contraste

La instrucción INK8 le indica al ordenador que el color que toma la tinta, en cualquier posición donde se incluya esta instrucción debe permanecer, no debe cambiar. La instrucción INK9 modifica el color de la tinta a blanco o negro, de forma que contraste con el color del fondo o color del papel.

Colores de tinta (INK) globales o locales

Cuando la instrucción está formada solamente por la palabra INK, como en el ejemplo anterior, el efecto es global (el color es global), y como resultado todas las demás imágenes se visualizarán con ese color de tinta. La palabra clave INK puede también insertarse en instrucciones que controlen la visualización de la información en la pantalla, y que incluyen alguna de las palabras claves siguientes: PRINT, INPUT, PLOT, DRAW y CIRCLE. INK se coloca a continuación de la palabra clave, pero precede la información o datos y los parámetros que controlan la visua-lización de la información. INK debe ir seguida por los mismos valores y un PUNTO Y COMA. Por ejemplo:

60 **CIRCLE INK** 4; 128,88,87

En este caso el efecto de la palabra clave INK es local y se aplica únicamente a los caracteres, puntos, líneas rectas o curvas ejec.-tadas por el ordenador en cumplimiento de las instrucciones contenidas en esa línea de programa La instrucción del ejemplo anterior le indica al ordenador que dibuje un circulo de color verde. Una vez se haya completado esta instrucción local, el ordenador retornará al color de tinta global o al color normal (negro).

Formato

INK expr número entero [;]

INKEY$ INput KEY

Posición en el teclado

MODO EXTENDIDO
N

Función

La palabra clave INKEY$ es utilizada para detectar si se ha oprimido alguna de las teclas en el teclado del ordenador.

Cómo emplear INKEY$

La palabra clave INKEY$ no requiere un argumento y por lo general es utilizada para asignar un carácter determinado a una variable de cadena, con el propósito de controlar un carácter en particular. Por ejemplo:

70 **LET** a$=**INKEYS**
130 IF **INKEY$**=«N»**THEN STOP**

Al ejecutarse la instrucción, INKEY$ retorna el valor del carácter producido por la tecla oprimida en ese momento. Si no se oprime una tecla, entonces INKEY$ devuelve una cadena vacía («»). Es importante distinguir entre mayúsculas y minúsculas y otros caracteres utilizados con o sin CAMBIO. Para detectar cualquier tecla sin tomar en cuenta el carácter específico producido por ella se utiliza la palabra clave IN.

A diferencia de la palabra clave INPUT. INKEY$ hace avanzar al ordenador inmediatamente a la instrucción siguiente. Por lo tanto, normalmente INKEY$ se coloca en el interior de un bucle que se repite constantemente hasta quena oprime la tecla deseada.

Ejemplo

Esta línea suspende la operación hasta que se introduce la letra «y», sin cambio a mayúsculas ni cierre de mayúsculas .

60 **IF INKEY$** <>«y» **THEN GOTO** 60

Formato

INKEY$

INPUT Entrada de datos

Posición en el teclado

I

Instrucción/Comando

La palabra clave INPUT permite introducir información o datos en el ordenador durante la ejecución de un programa.

Cómo emplear INPUT

Normalmente la palabra clave INPUT forma parte de una instrucción contenida en un programa y se utiliza de manera muy similar a PRINT. En su forma más sencilla, INPUT debe ir seguida por una variable numérica o de cadena, por ejemplo:

60 **INPUT** x
90 **INPUT** a$

Al llegar a esta instrucción el ordenador espera hasta que se introduzca algún número en él. A medida que se va introduciendo la info.-mación a través del teclado, el valor o dato se visualiza en á línea inferior de la pantalla Al oprimir la tecla ENTER el valor o dato introd.-cido se asigna a la variable indicada y el programa continúa.

Una instrucción INPUT puede contener más de una variable y puede visualizar mensajes, advirtiendo la necesidad de introducir ciertos datos o números. Este efecto se logra exactamente de la misma forma que con la palabra clave PRINT, utilizando comillas para delimitar el mensaje que pide la información, e incluyendo los puntos y comas necesarios para separar los diferentes elementos. Se pueden incluir instrucciones tales como INK, FLASH y PAPER en la instrucción que contiene INPUT. Por ejemplo:

80 **INPUT INK**2 «¿Como te llamas?», n$, («¿Cuántos años tienes,»+n$+«?»); edad

Tome nota de las siguientes diferencias en-tre las instrucciones que contienen INPUT y PRINT. INPUT espera cuando llega a una variable, por lo tanto todas las variables y expresiones (por ejemplo la variable que incluye el signo $ en el ejemplo) que se incluyan en la pregunta deben IR ENTRE PARENTESIS. El ordenador visualiza el mensaje al pie de la pantalla, y las líneas ascienden a medida que se van agregando a nuevos elementos. La palabra clave AT se puede incluir en una instruíción que incluye INPUT, como ocurría en instrucciones con PRINT. La palabra clave AT 0,0 visualiza el mensaje al principio de la línea, encima de la última línea de la pantalla. Si se visualizan más de dos líneas la imagen ascenderá en la pantalla.

Cómo interrumpir INPUT

Cuando INPUT va seguido de una variable numérica y se introduce la palabra clave STOP, el programa se detiene. En el caso de variables de cadena, para interrumpir la operación de á palabra clave INPUT, se sigue el procedimiento siguiente: borrar el primer par de comillas que aparezca en la pantalla, y a continuación introducir la palabra clave STOP.

EMPLEO DE INPUT CON LINE

La instrucción INPUT LINE sólo puede utilizarse con variables de cadena. Normalmente la combinación de INPUT con una variable de cadena hace que el ordenador visualice dos

pares de comillas en la pantalla. A medida que se introduce la cadena en el ordenador mediante el teclado, su contenido va apareciendo en la pantalla entre comillas. Para borrar las comillas utilice la instrucción INPUT LINE, seguida de la variable de cadena. Si es necesario enviar un mensaje al usuario para que introduzca información, el mensaje se sitúa entre las palabras clave INPUT y LINE. Por ejemplo:

70 **INPUT** «¿Cómo te llamas?»; **LINE**n$

Formato

INPUT [pregunta][;][,][']
var-num
INPUT [pregunta][;][,][']
var de cadena
INPUT [pregunta][;][,]['] LINE
var de cadena
[pregunta] = [const de cadena] [(expr de cadena)][**AT** expr número entero, expr número entero][sentencia][;][,][']

INT INTeger — Número Entero

Posición en el teclado

MODO EXTENDIDO
R

Función

La palabra clave INT transforma números fraccionarios en números enteros suprimiendo la parte fraccionaria.

Cómo emplear INT

La palabra clave INT debe ir seguida por un valor numérico, por ejemplo:

70 **LET** x = **INT** y

Las expresiones deben ir ENTRE PARENTESIS. INT produce el número fraccionario redondeándolo POR DEBAJO (en menos) al número entero más próximo.

Ejemplo:

La siguiente sentencia contiene dos números fraccionarios, uno de ellos negativo. Al introducirse el comando PRINT.

PRINT INT 45.47,**INT** -7.76

el ordenador visualizará lo siguiente en la pantalla

45 8

Formato

INT const-num
INT var-num
INT (expr-num)INVERSE Inverso

INVERSE — Inverso

Posición en el teclado

MODO EXTENDIDO
M

Instrucción/Comando

La palabra clave INVERSE produce el efecto de invertir los colores en las posiciones de los caracteres, con el resultado de que los colores de tinta se tornan colores del papel y viceversa.

Cómo emplear INVERSE

Normalmente INVERSE se utiliza para formar una instrucción incluida en un programa. INVERSE va seguido por un valor numérico. Por ejemplo:

70 **INVERSE** 1

El valor numérico a continuación de la palabra clave INVERSE se redondea al número entero más próximo, que puede ser 0 ó 1. El efecto de INVERSE 1 es hacer que todas las imágenes visualizadas en ejecución de las palabras clave PRINT y OUTPUT se produzcan en colores invertidos. Por el contrario, INVERSE 0 restituye el color normal de papel y tinta (fondo y primer plano).

Observe que la palabra clave INVERSE puede incluirse en el interior de una instrucción que controle la visualización de información en la pantalla, como sucede por ejemplo con la palabra clave INK. INVERSE puede causar problemas cuando se utiliza conjuntamente con CIRCLE, PLOT o DRAW. En este caso el efecto de la instrucción INVERSE 1 será que el punto, línea o círculo se tracen en el mismo color que el papel o fondo de la imagen, y en consecuencia, éstos desaparecerán.

Formato

INVERSE expr número entero

LEN LENgth — Longitud

Posición en el teclado

MODO EXTENDIDO
K

Función

La palabra clave indica el largo o extensión de una cadena (string).

Cómo emplear LEN

LEN debe ir seguida por un valor de cadena, por ejemplo:

50 **LET** x = **LEN** a$

Las expresiones deben ir ENTRE PARENTESIS. LEN informa del número de caracteres de la cadena.

Ejemplo

La siguiente línea

```
120 INPUT a$:IF LEN a$>9
    THEN GOTO 120
```

pasa sólo cadenas que contengan más de nueve caracteres.

Formato

LEN const de cadena
LEN var de cadena
LEN (expr de cadena)

LET

Posición en el teclado

L

Instrucción/Comando

La palabra clave LET se utiliza para asignar un valor a una variable. En el dialecto de BASIC que utiliza el INVES SPECTRUM+ siempre es necesario incluir LET en las instrucciones que asignan valores a variables.

Cómo emplear LET

Normalmente, LET forma parte de una instrucción incluida en un programa aunque también puede utilizarse como comando directo. Esta palabra clave debe ir seguida por una variable de cadena o una variable numérica, un signo igual y a continuación un valor. Este valor puede ser numérico o de cadena, dependiendo de la naturaleza de la variable que acompañe a LET. Por ejemplo:

```
60 LET x=x+ 1
80 LET a$=«Correcto»
```

Este valor es asignado entonces a la variable indicada (x en el primer ejemplo y a$ en el segundo). Las variables simples (son variables sin subíndices) se mantienen indefinidas hasta que se les asigna un valor mediante sentencias que contengan las palabras clave LET, READ o INPUT. Las variables contenidas en matrices (arrays) reciben inicialmente los valores 0 o cadena vacía (« »). (Ver DIM).

Formato

LET var-num = expr-num
LET var de cadena = expr de cadena

LINE — Línea

Posición en el teclado

MODO EXTENDIDO
CAMBIO 3

Ver INPUT, SAVE.

LIST — Lista

Posición en el teclado

K

Comando/Instrucción

Al ejecutar la instrucción LIST el ordenador visualiza en la pantalla una lista del programa contenido en su memoria en ese momento.

Cómo emplear LIST

Normalmente LIST se utiliza como comando directo, aunque también puede formar parte de una instrucción dentro de un programa. Cuando lo que se desea es hacer una lista del contenido de un programa completo, basta con introducir la palabra clave LIST sin más información. Después del comando directo.

```
LIST
```

el ordenador visualiza en la pantalla la primera página y siguientes, con las líneas del programa. Este proceso se cumple por páginas. Para desplazar las páginas hacia arriba (scroll), basta con oprimir cualquier tecla excepto N, la barra espaciadora, STOP o PARAR.

LIST puede también ir seguido de un número de línea, determinado (un valor numérico), por ejemplo:

```
LIST 100
```

si es necesario, el valor numérico a continuación de LIST se redondea al número entero más próximo. A continuación el ordenador visualiza una lista comenzando a partir de esa línea. Si no existe una línea con ese número, el ordenador comenzará su lista en la próxima línea.

Formato

LIST (expr número entero)

LLIST

Posición en el teclado

MODO EXTENDIDO
V

Comando/Instrucción

Esta palabra clave le indica a la impresora que imprima una copia de la lista del programa contenido en la memoria del ordenador en ese momento.

Cómo emplear LLIST

LLIST se utiliza exactamente de la misma forma que LIST (ver LIST por más detalles). Observe que la información representada en la pantalla no cambiará mientras la lista se esté imprimiendo.

Formato

LLIST (exp. número entero)

LN Logarithm (Natural) — Logaritmo (Natural)

Posición en el teclado

MODO EXTENDIDO
Z

Función

LN suministra el logaritmo natural (en base e) de un valor especificado. Actúa como el inverso de EXP.

Cómo emplear LN

La palabra clave LN va seguida por un valor numérico. Por ejemplo:

60 **LET** x=**LN** y

Las expresiónes deben ir ENTRE PARENTESIS. El valor numérico a continuación de LN debe ser mayor que 0. LN producirá el logaritmo natural de ese valor.

Formato

LN const-num
LN var-num
LN (expr-num)

LOAD — Cargar

Posición en el teclado

J

Comando/Instrucción

La palabra clave LOAD le indica al ordenador que cargue un programa completo almacenado en cinta, depositándolo en su memoria.

Cómo emplear LOAD

Normalmente LOAD es utilizado como un comando directo, pero también puede formar parte de un programa, para cargar otro programa.

La palabra clave LOAD va seguida por el nombre del programa que se va a cargar. Los nombres de los programas son valores de cadena con una extensión máxima de diez caracteres. Por ejemplo:

LOAD «nombre»

Los efectos de LOAD son drásticos: al ejecutarse esta instrucción, el ordenador borra el programa que está en ese momento en su memoria, y también todos los valores de sus variables. A continuación, el INVES SPECTRUM+ busca en la cinta el nombre del nuevo programa y lo carga.

Recuerde que en el caso de los nombres de programas el ordenador distingue entre letras mayúsculas y minúsculas.

Cuando LOAD va seguido por una cadena vacía, como por ejemplo:

LOAD “”

El INVES SPECTRUM+ carga el primer programa completo que encuentre.

Formato

LOAD expr de cadena

LOAD CODE

Posición en el teclado

MODO EXTENDIDO
I

Comando/Instrucción

La palabra LOAD CODE se emplea para indicarle al ordenador que cargue en su memoria cierta información que ha sido almacenada previamente en cinta.

Esta información consiste en una serie de bytes que se enviarán a diversas direcciones de la memoria del ordenador.

LOAD también puede usarse para cargar una imagen gráfica u otro tipo de información de gráficos definidos por el usuario.

Cómo emplear LOAD CODE

LOAD CODE puede utilizarse como comando directo o alternativamente como parte de una instrucción de un programa.

La sintaxis de esta palabra clave es la siguiente: en primer lugar LOAD, luego el nombre del programa entre comillas, finalmente CODE. Por ejemplo:

LOAD «datos» **CODE**

El nombre de programa que sigue a LOAD es el nombre de la información registrada en la cinta, que se desea cargar al ordenador. A este nombre se le aplican las mismas limitaciones que en el caso de la palabra clave LOAD (véase LOAD).

Cuando el ordenador llega a esta instrucción en el programa —o cuando recibe el comando directo— procede a buscar en la cinta la información almacenada bajo ese nombre. Cuando el ordenador encuentra la información, visualizará los bytes, seguidos por el nombre de ese grupo de datos.

A continuación, el INVES SPECTRUM+ carga los bytes en su memoria distribuyéndolos en las direcciones a partir de las cuales se registraron anteriormente. Esta nueva info.-mación cargada por la ejecución del comando se superpone sobre cualquier otra información anterior que se encontrará en la memoria del ordenador.

CODE puede también ir seguido por uno o dos valores numéricos separados por una coma.

LOAD «dibujo» **CODE** 16384,6912

Los valores numéricos incluidos a continuación de CODE se redondean si es necesario al número entero más cercano. Su propósito es definir la dirección inicial desde donde se cargará la información indicada en la instruíción (16384 en el ejemplo), y también indicar la cantidad y bytes que se enviarán a las direcciones comenzando a partir de esa dirección.

Si el número indicado no es el correcto, el ordenador visualizará el mensaje de error de carga de la cinta (ERROR DE CARGA).

Si LOAD va seguido de un único valor unmérico, el ordenador interpreta que se trata del número que define la posición de la dirección inicial a partir de la cual deben almacenarse todos los bytes.

También se puede obtener el mismo resoltado que en el ejemplo anterior utilizando las palabras clave LOAD SCREEN$.

Para más detalles sobre el procedimiento empleado para almacenar bytes, véase la sección SAVE CODE.

Formato

LOAD expr de cadena CODE [expr entero] [,expr número entero]

LOAD DATA

Posición en el teclado

MODO EXTENDIDO
D

Instrucción/Comando

La palabra clave LOAD DATA se emplea para cargar matrices (arrays) almacenadas en cinta. Las matrices se registran en cinta mediante una instrucción SAVE DATA.

Cómo emplear LOAD DATA

Las palabras clave LOAD DATA pueden emplearse bien como parte de una instrucción de un programa o como un comando directo. El orden seguido es el siguiente: a continuación de LOAD se incluye el nombre que identifica la información registrada en la cinta (un valor de cadena). A continuación se incluye la palabra clave DATA y una letra o una letra y un signo de $. Los últimos elementos de la instrucción son un par de paréntesis vacíos. Por ejemplo:

270 **LOAD** «números» **DATA** n( )
300 **LOAD** «nombres» **DATA** N$( )

El nombre que se incluye a continuación de LOAD es el nombre de la matriz registrada en la cinta. Este nombre se encuentra sujeto a las mismas restricciones que los demás nombres de programas, cuando se utilizaba la palabra clave LOAD. La letra o letra$ colocada a continuación de DATA es el nombre de letra seleccionada para designar la matriz en el programa contenido en la memoria del INVES SPECTRUM+, y que será utilizada en su momento por el ordenador.

Al ejecutarse esta instrucción, el INVES SPECTRUM+ busca la matriz indicada en la cinta. Cuando la encuentra, el ordenador visualiza el mensaje "Matriz Num.:" o "Matriz Caract.:" en la pantalla, seguido del nombre. A continuación el ordenador carga la matriz en su memoria.

Cuando el ordenador carga la matriz registrada en la cinta, la deposita en su memoria y borra cualquier matriz con el mismo nombre (la misma letra) que se encuentre con anterioridad en su memoria (en el ejemplo, los nombres de las matrices son «n» y «n$»); finalmente, el ordenador forma una nueva matriz designándola con el nombre de letra especificado.

Formato

LOAD expr de cadena **DATA** letra [$]( )

LOAD SCREEN$

Posición en el teclado

J
MODO EXTENDIDO
CAMBIO K
K

Instrucción/Comando

LOAD SCREEN$ le indica al ordenador que cargue la información referente a una imagen de la pantalla de televisión que se encuentra almacenada en cinta.

Como resultado de esta instrucción, el ordenador toma la información de la cinta y la envía a su memoria (a la parte de ésta que controla la visualización de imágenes en la pantalla de televisión), para generar la imagen.

Cómo emplear LOAD SCREEN

LOAD SCREEN$ puede integrarse en una instrucción incluida dentro de un programa, o bien puede utilizarse como comando directo.

LOAD va seguido del nombre bajo el cual se registró la información en la cinta. Este nombre es un valor de cadena. Finalmente se agrega SCREEN$. Por ejemplo:

LOAD «dibujo» **SCREEN$**

El nombre entre paréntesis a continuación de LOAD es el nombre que identifica la info.-mación almacenada en la cinta, y se encuentra sometido a las mismas restricciones que los nombres de programas cuando se utiliza la palabra clave LOAD.

La imagen se forma gradualmente en la pantalla con los colores de tinta y papel ac-

tivos en ese momento y luego toma los colores definidos como atributos.
Para más detalles acerca del procedimiento para almacenar la información contenida en la pantalla, ver SAVE SCREEN$.

Formato

LOAD expr de cadena **SCREEN$**

LPRINT Line printer PRINT

Posición en el teclado

MODO EXTENDIDO
C

Instrucción/Comando

La palabra clave LPRINT le indica a la impresora que imprima una determinada info.-mación de la misma forma que la palabra clave PRINT le indica al ordenador que visualice una determinada información en la pantalla de la televisión.

Cómo emplear LPRINT

LPRINT puede formar parte de una instruíción, de un programa, o bien puede utilizarse como comando directo. Debe ir seguido por elementos de información o datos que pueden separarse por puntos y comas o apóstrofes, por ejemplo:

60 **LPRINT** "Número»;x, «Nombre»; n$, «Edad»,a

Al ejecutar esta instrucción, el ordenador transfiere la información indicada a la impresora, que lo imprime. El formato será el mismo que se hubiera utilizado para visualizar la información en la pantalla al ejecutar la instruíción PRINT.
Una instrucción que contenga LPRINT también puede incluir instrucciones TAB, algunos controles CHRS, e instrucciones INVERSE y OVER. También puede incluir códigos de control, con los mismos resultados que cuando se utiliza PRINT.

Formato

LPRINT [**TAB** (expr número entero;)] [**AT** expr número entero, expr número entero;] [**CHR$** (expr número entero);] [sentencia;] [expr número entero] [expr de cadena] [;] [,] ['].

MERGE Mezclar

Posición en el teclado

MODO EXTENDIDO
CAMBIO T

Instrucción/Comando

La palabra clave MERGE permite fusionar dos programas.

Cómo emplear MERGE

MERGE se puede incluir en una instrucción de un programa o se puede utilizar como comando directo.
MERGE debe ir seguida por el nombre de la información registrada en cinta como un valor de cadena, por ejemplo:

500 **MERGE** «prog2»

El nombre a continuación de MERGE («prog2» en el ejemplo) es el nombre del programa almacenado en cinta que se desea combinar con el programa que ya se encuentra en la memoria del ordenador.
Este nombre se encuentra sometido a las mismas limitaciones que los nombres de programas utilizados en sentencias con las palbras claves LOAD.
El ordenador ejecuta la instrucción MERGE, cargando el programa almacenado en cinta sin borrar el otro programa en su memoria.

Formato

MERGE expr de cadena

MOVE

Comando utilizado para controlar el almacenamiento de información en los Microdrives. Para mayor información, consúltese el Manual de Microdrives e Interface 1.

NEW

Posición en el teclado

A

Comando/Instrucción

Cuando se introduce la palabra clave NEW, el ordenador borra el contenido del área de BASIC de la memoria (la zona que se extiende hasta el valor límite RAMTOP) eliminando todos los programas que se encuentran en esa área de la memoria.

Cómo emplear NEW

Por lo general, NEW se utiliza como comando directo, pero también puede formar parte de una instrucción dentro de un programa. Se utiliza sólo (sin parámetros).
Al ejecutar esta instrucción, el ordenador borra el programa y las variables de su memoria hasta el valor límite RAMTOP. Como resultado, los gráficos definidos por el usuario y conservados en un área por encima del límite RAMTOP no se ven afectados por NEW.

Formato

NEW

NEXT Siguiente

Posición en el teclado

N

Instrucción/Comando

La palabra clave NEXT se utiliza siempre en conjunción con FOR para construir un bucle FOR NEXT.

Cómo emplear NEXT

NEXT se utiliza normalmente en una instruíción de programa y como parte de un bucle FOR NEXT. Va seguido de una letra, que es la variable de control del bucle. Por ejemplo:

90 **NEXT** a

En el dialecto de BASIC que emplea el INVES SPECTRUM+ la variable de control debe especificarse siempre.

Para más detalles sobre los bucles (loops) FOR NEXT, véase la sección sobre FOR NEXT.

Formato

NEXT letra.

NOT No

Posición en el teclado

CAMBIO
S

Operador lógico/función

La palabra clave NOT se emplea para invertir la veracidad de una condición, causando que una condición falsa se torne verdadera y viceversa.

Cómo emplear NOT

NOT va seguido por una condición o un valor numérico. Dos ejemplos:

90 **IF NOT** x=y **THEN PRINT** «Error»
90 **LET** correcto=x=y+z: **IF NOT** correcto **THEN PRINT** «Error»

Cuando NOT va seguido por una condición (x=y+z en el ejemplo), el INVES SPECTRUM+ opera del siguiente modo: primero asigna un valor 1 a la condición si es verdadera, y 0 si es falsa. NOT opera entonces como función, invirtiendo el valor producido obteniendo como resultado la posibilidad de examinar la validez de la condición contraria.

Observe que si la condición contiene las palabras clave AND u OR debe *estar encerrada entre paréntesis*.

Cuando NOT vaya seguido por un valor unmérico, devolverá un 0 si el valor numérico que le sigue no es 0, y un número 1 si el valor que le sigue es 0. Por lo tanto, en los ejemplos anteriores ocurrirá lo siguiente: el INVES SPECTRUM+ visualizará «Error» si x<>y+z, o «correcto» si tiene un valor de 0.

Formato

NOT cond
NOT expr-num

OPEN

Comando utilizado para controlar el almacenamiento de información en los Microdrives. Para más información consulte el Manual de Microdrives e Interface 1.

OR

Posición en el teclado

CAMBIO U

Operador lógico/función

OR actúa como un operador lógico para establecer la veracidad de una combinación de condiciones. Si una o más condiciones son verdaderas, la condición total es verdadera. OH también opera como una función para ejecutar operaciones binarias que actúen sobre dos valores numéricos.

Como emplear OR

Como un operador lógico. OH enlaza dos condiciones en una instrucción donde se trata de establecer la veracidad o falsedad del conjunto total. Por ejemplo:

70 **IF INKEY$=** «N» **OR INKEY$=**«n» **THEN STOP**

Si cualquiera de las dos condiciones indicadas en la instrucción es verdadera, entonces el ordenador llega a la conclusión de que la totalidad de la instrucción es verdadera y obedece la instrucción (THEN STOP).

En el ejemplo una de las condiciones (INKEY$=«N» o INKEY$=«n») se transforma en verdadera tan pronto como se oprime la tecla «N» en el teclado del ordenador, indiferentemente de que se trate de la «N» mayúscula o minúscula. En los dos casos (mayúscula o minúscula) la totalidad de la combinación en la instrucción se ha transformado en verdadera y el programa se detiene obedeciendo la instrucción THEN STOP.

OR como una función

El INVES SPECTRUM+ asigna un valor de 1 a la condición verdadera y 0 a la condición falsa. La palabra clave reconoce cualquier valor excepto 0 como valor verdadero, y 0 como valor falso. Por tanto, OH puede ir precedido o seguido por un valor numérico, por ejemplo:

40 **LET** x=y **OR** z

En el ejemplo, el ordenador asignará un valor de 1 a la variable x si el valor de z es cualquier número excepto 0. Alternativamente, si el valor de z es 0 o una condición falsa, el ordenador le asignará un valor y.

Recuerde que el INVES SPECTRUM+ no evalúa combinaciones de valores numéricos de acuerdo con las tablas de veracidad estándar.

Formato

cond **OR** con
expr-num **OR** expr-num.

OUT — Salida

Posición en el teclado

MODO EXTENDIDO
CAMBIO O

Instrucción/Comando

La palabra clave OUT envía un byte a la dirección de una «puerta», o punto de entrada/salida en el ordenador.

Cómo emplear OUT

OUT puede emplearse para formar una instrucción dentro de un programa o como comando directo, independiente. OUT debe ir seguido por dos valores numéricos SEPARADOS POR UNA COMA. Por ejemplo.

40 **OUT** 254,3

Ambos valores se redondean al número entero más próximo. El primer valor (254 en el ejemplo) puede variar entre 0 y 65535 y representa la dirección de la «puerta» o punto de conexión a donde se envían los datos. El segundo valor (3 en el ejemplo) puede variar entre 0 y 255 y es el byte que se envía a la dirección especificada con anterioridad.

Como se explicó, cada byte se compone de una serie de bits. En el ejemplo los bits 0-2 del byte enviado a la dirección de la puerta 254 definen el color del borde o bordes. En el ejemplo, esta instrucción cambiará el color del borde a magenta (morado). El bit número 3 enviado a la misma dirección controla el enchufe MIC y el bit 4 controla el altavoz.

La dirección de «port», o punto de con-tacto, 251 controla la impresora, y las direc-ciones de punto de contacto 254, 247, y 239 son utilizadas para controlar otros equipos periféricos conectados al ordenador.

Formato

OUT expr número entero, expr número entero.

OVER — Sobreimpresión

Posición en el teclado

MODO EXTENDIDO
CAMBIO N

Instrucción/Comando

La palabra clave OVER es utilizada para sobreimprimir un carácter encima de otro. También puede ser empleada para dibujar puntos o trazar líneas rectas o curvas de color de pa-pel (paper) en lugar de color de tinta (ink).

Cómo emplear OVER

Por lo general OVER se utiliza para formar una instrucción, dentro de un programa. Va seguida por un valor numérico. Por ejemplo:

80 **OVER** 1

El valor numérico a continuación de la palabra clave OVER es redondeado al número entero más próximo, y puede ser 0 ó 1. El valor normal es OVER 0, y determina que un nuevo carácter impreso en una posición de carácter borre completamente lo que había antes en la misma posición, tomando su lugar. En cambio el valor OVER 1 pone en operación un efecto especial: el nuevo carácter se agregará al carácter anterior en la misma posición, combinándose con él.

OVER puede insertarse en una instrucción que contenga las palabras clave PRINT o INPUT, como sucede con INK. En este caso, su efecto es local, afecta únicamente a los caracteres que se visualicen como resultado de la ejecución de esa instrucción. Por ejemplo, la instrucción siguiente producirá el efecto de subrayar una palabra.

60 **PRINT AT** 11,15; ,,SI»;
OVER 1; **AT** 11,15

Sin embargo, observe que los caracteres se combinan de tal forma que, en el sector donde los dos caracteres se sobreponen, el ordenador visualiza el color del papel (paper).

OVER en el modo de alta resolución

OVER puede ser utilizado conjuntamente con PLOT, DRAW y CIRCLE. Cuando no se incluye la palabra clave OVER las líneas rectas y curvas pueden sobreponerse las unas a las otras, pero deben ser del mismo color de tinta, para evitar que cambie de color de la totalidad de la posición donde ambas líneas se cruzan. En cambio, si se incluye la palabra clave OVER 1, cuando dos líneas rectas o curvas se cruzan, o cuando se encuentran con algún otro carácter, se visualizará el color del papel. Otro efecto interesante es que si se dibujan puntos, o se trazan dos líneas rectas o curvas, exactamente en las mismas posiciones, en presencia de la instrucción OVER 1, los elementos sobrepuestos desaparecerán.

Formato

OVER exp. número entero

PAPER — Papel

Posición en el teclado

MODO EXTENDIDO
CAMBIO C

Instrucción/Comando

PAPER es utilizado para seleccionar el color del papel o fondo de la pantalla. Este color puede ser el de la totalidad del fondo, o solamente del fondo de ciertos caracteres, puntos o líneas visualizados en las posiciones de caracteres individuales.

Cómo emplear PAPER

La palabra clave PAPER puede utilizarse para formar una instrucción dentro de un programa o como comando directo. PAPER va seguido por un valor numérico, por ejemplo.

80 **PAPER** x

Este valor numérico se redondea al número entero más próximo y puede variar entre 0 y 9. Los números de los códigos de los colores de PAPER son similares a los que se utilizan con la palabra clave INK (véase la sección correspondiente). Como en el caso de INK, los colores pueden ser globales o locales. Para hacerlos locales, la palabra clave PAPER se inserta en instrucciones que controlen la visualización de imágenes en la pantalla, exactamente de la misma manera que con los colores de tinta (véase la sección INK).

Cuando se visualizan en la pantalla caracteres a continuación de una instrucción que incluya la palabra clave PAPER (independientemente de que la instrucción tenga un efecto global o local), el color del fondo de la posición del carácter correspondiente cambia al color seleccionado. Esto también es aplicable cuando se dibujan puntos o se trazan líneas rectas o curvas mediante instrucciones que contengan el comando PAPER (efecto local), pero no a continuación de una instrucción o comando de efecto global.

Para producir un fondo de color que abarque la totalidad del área de visualización en la pantalla, es necesario utilizar la palabra clave CLS después de una instrucción que con-tenga PAPER. Como resultado, toda la imagen de la pantalla se torna del color indicado, permaneciendo como color de fondo.

Formato

PAPER expr número entero [;]

PAUSE — Pausa

Posición en el teclado

M

Instrucción/Comando

El efecto de la palabra clave PAUSE es suspender la ejecución del programa por un período de tiempo definido o indefinido.

Cómo emplear PAUSE

La palabra clave PAUSE se utiliza normalmente como parte de una instrucción dentro de un programa. Va seguida por un valor unmérico, por ejemplo:

130 **PAUSE** 100

El valor numérico situado a continuación de PAUSE se redondea al número entero más próximo. El número puede variar entre 0 y 65535, y define la duración de la pausa en términos de imágenes de televisión (50 imágenes por segundo). Por ejemplo, un valor de 50 produce una pausa de un segundo, donde la frecuencia de las imágenes es de 50 Hz.

Sin embargo, la pausa puede interrumpirse presionando una tecla. La instrucción PAUSE 0 crea una pausa de duración ilimitada, que continúa hasta que se interrumpe oprimiendo una tecla.

Formato

PAUSE expr número entero.

PEEK — Mirar

Posición en el teclado

MODO EXTENDIDO
O

Función

PEEK suministra el valor del byte almacenado en una dirección particular de la memoria del ordenador.

Cómo emplear PEEK

La palabra clave PEEK va seguida por un valor numérico, por ejemplo:

80 **LET** x = **PEEK** (256*y)

Observe que la expresión debe ir ENTRE PARENTESIS. El valor numérico que se en-- cuentra a continuación de PEEK se redondea si es necesario al número entero más próximo. El número puede variar entre 0 y 65535, e identifica una determinada dirección de la memoria. Como resultado de esta instrucción, el ordenador suministra el valor del byte (un número 0 y 255) en la dirección especificada.

Ejemplo

El número de imágenes (pantallazos) de televisión que se han producido desde el mo-

mento en que se conectó el INVES SPECTRUM+ se almacena en las direcciones 23672 a 23674. Como esas imágenes se producen a una velocidad constante, podemos explorar estas posiciones de memoria mediante una instrucción que contenga la palabra clave PEEK, y habremos obtenido un método preciso para medir el tiempo. La línea de programa que se incluye en el ejemplo siguiente hace que el ordenador especifique el tiempo transcurrido desde que se conectó el INVES SPECTRUM+ (en segundos).

10 **PRINT** (**PEEK** 23672+256* **PEEK** 23673+65536***PEEK** 23674)/50

Formato

PEEK const número entero
PEEK var número entero
PEEK (expres número entero)

PI — *π*

Posición en el teclado

MODO EXTENDIDO
M

Función

PI suministra el valor de pi (*π*) para su utilización en cálculos.

Cómo emplear PI

Si esta palabra clave no requiere valores o variables complementarias sí se utiliza en instrucciones o comandos, por ejemplo:

DRAW 25,0,-PI

PI suministra un valor de 3.1415927, y como resultado el comando en el ejemplo le indicará al ordenador que trace un semicírculo de gran tamaño en la pantalla.

Formato

PI

PLOT — Dibujar

Posición en el teclado

Q

Instrucción/Comando

PLOT se utiliza para dibujar un pixel o punto de color de una determinada posición de la pantalla, en la producción de gráficos de alta resolución.

Cómo emplear PLOT

La palabra clave PLOT puede formar parte de una instrucción incluida en un programa o también puede utilizarse como comando. Normalmente va seguida por dos valores numéricos separados por una coma, por ejemplo:

50 **PLOT** 128,87

Los dos valores numéricos después de PLOT se redondearán al número entero más próximo si fuera necesario. El primer valor puede variar en 0 y 255, su función es definir la coordenada horizontal de la posición en la pantalla. El segundo valor puede variar entre 0 y 175, su función es definir la coordenada vertical de la posición. Normalmente el ordenador ejecuta las instrucciones dibujando un pixel en el punto indicado, utilizando el color de tinta efectivo en ese momento.

Tome nota de los siguientes efectos, que se producen sobre la instrucción PLOT debido a otras instrucciones o comandos que afectan a los colores. Después de OVER 1, el color de un punto que se encuentra en la misma posición cambia a color papel (paper). A continuación de INVERSE 1, el punto se dibuja en el mismo color de papel vigente en ese momento. Después de BRIGHT 1 o FLASH 1, la posición completa de un carácter donde se dibuja el pixel en la pantalla de baja resolución, tomará un color brillante o relampagueará.

Estas cuatro palabras clave e INK también pueden insertarse en una instrucción que contenga la palabra clave PLOT. El procedimiento a seguir es el mismo que se utilizó en el caso de la palabra clave PRINT.

160 **PLOT INK** 2;x,y

El efecto de estas palabras claves es el mismo que antes, con una diferencia importante: será un efecto limitado al pixel dibujado en ejecución de la sentencia. Cuando se inserta la palabra clave PAPER en una sentencia conteniendo PLOT, el efecto de la instruíción será cambiar el color de papel de toda la posición del carácter donde se encuentra el papel.

Formato

PLOT [sentencia;] expr número entero, expr número entero

POINT — Punto

Posición en el teclado

MODO EXTENDIDO
CAMBIO 8

Función

La palabra clave POINT se emplea para averiguar si el color de una determinada posición de la pantalla de alta resolución es un color de tinta o es un color de papel.

Cómo emplear POINT

POINT va seguido por dos valores numéricos separados por una coma y encerrados entre paréntesis. Por ejemplo:

240 **IF POINT** (x,y) = 1 **THEN GOSUB** 600

Los dos valores situados después de POINT se redondean al número entero más próximo, si fuera necesario. La función del primer valor es definir la coordenada horizontal de un pixel en la pantalla. Puede variar entre o y 255. El segundo valor define la coordenada vertical y puede variar entre 0 y 175. Al recibir la instrucción POINT, el ordenador responderá con un número 1 si el pixel de la posición indicada es de color tinta (ink), y con un 0 si es color papel (paper).

Formato

POINT (expr número entero, expr número entero)

POKE Introducir

Posición en el teclado

O

Instrucción/Comando

La palabra clave POKE se utiliza para cambiar el valor de un byte situado en una dirección específica de la memoria del ordenador.

Cómo emplear POKE

POKE puede formar parte de una instruíción incluida en un programa o también puede operar como comando. Va seguido por dos valores numéricos *separados por una coma*. Por ejemplo:

POKE 23609,255

Los dos valores a continuación de POKE se redondean al número entero más próximo si es necesario. El primer número es una dirección de la memoria RAM del ordenador que puede variar entre 16384 y 65535. El segundo valor puede variar entre -255 y 255, y es el byte cuyo valor se desea modificar.

Formato

POKE expr número entero, expr número entero.

PRINT imprimir

Posición en el teclado

P

Instrucción/Comando

La instrucción que contiene PRINT puede llevar incorporadas además otras palabras clave para definir la posición y color de la información visualizada.

Cómo emplear PRINT

PRINT puede emplearse independientemente o puede ir seguida por datos. Estos datos pueden ser cualquier tipo de expresión numérica o de cadena o una combinación de ambos tipos.

Cuando se emplea PRINT con datos, los diferentes elementos deben separarse por un *punto* y *coma*, *una coma* o *un apóstrofe*.

También es posible insertar ciertas palabras claves entre PRINT y los datos, en cualquier orden; siempre y cuando la sentencia formada por aquella palabra clave *termine* en un punto y coma. Estas palabras claves son las siguientes: CHR$, TAB, AT, INK, PAPER, FLASH, BRIGHT, INVERSE y OVER.

Empleo de PRINT con cadenas

El efecto de la palabra clave PRINT sola, o seguida por una cadena vacía («»), será visualizar, «imprimir», una línea blanca en la pantalla y desplazar el cursor al principio de la línea siguiente.

PRINT seguida por una constante de cadena tendrá el efecto de visualizar en la pantalla los caracteres que aparecen entre comillas. El comando:

PRINT «3/542/76/21»

producirá el resultado siguiente en la pantalla:

3/542/76/21

PRINT seguida por una variable o expresión de cadena visualizará o «imprimirá» en la pantalla la cadena.

PRINT con números

PRINT seguido por cualquier variable numérica visualizará en la pantalla el valor de la expresión. Los números estarán en notación decimal con un total de hasta ocho dígitos significativos.

Los números muy grandes y muy pequeños se visualizan en notación científica (dos cifras separadas por la letra E). En esta notación, la primera parte (la mantisa) se multiplica por 10 elevado a la segunda parte (el exponente). El comando:

PRINT 3/542/76/21

por ejemplo, producirá el siguiente resultado en la pantalla

3.4680798E-6

Impresos con PRINT mediante los signos de puntuación

La información «impresa» en la pantalla en ejecución de una instrucción PRINT seguida por elementos de información, datos separados por un punto y coma, visualiza dichos elementos uno junto al otro. El comando:

PRINT 1;2;3

producirá

123

PRINT va seguida por elementos de información separados por comas, y el ordenador

visualizará cada uno de esos elementos en el centro o al principio de una línea de la pantalla, dependiendo de la posición del primer elemento de información. El comando

PRINT 1,2,3

producirá el siguiente resultado:

```
1                              2
3
```

PRINT seguida por elementos de información separados por apóstrofes visualizará cada uno de dichos elementos al principio de la línea siguiente. El apóstrofe opera como un «punto y aparte». El comando

PRINT 1'2'3

Producirá el siguiente resultado:

```
1
2
3
```

PRINT termina en un punto y coma, coma o apóstrofe, y la información visualizada por el ordendor cuando ejecute la próxima instruíción PRINT se verá afectada del mismo modo.

PRINT y otras palabras claves

PRINT puede ir seguido por TAB, un valor numérico, punto y coma y un elemento de infomación, por ejemplo:

60 **PRINT TAB** x;a$

El valor que aparece a continuación de TAB (x en el ejemplo) se redondeará, si es necesario, al número entero más próximo. A continuación, ese número se divide por 32, y se toma el resto de la división. El resultado es un valor entre 0 y 31.

PRINT puede ir seguido por las palabra clave AT y dos valores numéricos separados por *una coma*, *un punto y coma* y *un elemento* de información. Por ejemplo.

50 **PRINT AT** l,c; «Información»

El primer valor (la variable l en el ejemplo) define la línea horizontal en la pantalla donde se «imprimirá» o visualizará la información. Este valor puede variar entre 0 y 21. El segundo valor, la variable c en el ejemplo, define el número de la columna donde se imprmirá el primer carácter o dígito de la información. El número de columna puede variar entre 0 y 31. El comando PRINT Al 11, 16," " visualiza una estrella en el centro de la pantalla.

PRINT puede ir también seguido por una o más funciones CHR$. Para más detalles consulte la sección sobre CHR$.

PRINT y las palabras claves que controlan los colores

La imagen que aparece en PRINT en la pantalla de televisión se ve afectada por las instrucciones y comandos relacionados con el color y que contengan la palabra clave INK PAPER, FLASH, BRIGHT, INVERSE y OVER que se puedan encontrar en las líneas anteriores del programa. Instrucción con PRINT y terminada en punto y coma; palabra clave seguida por un punto y coma y a continuación el elemento de información, por ejemplo:

50 **PRINT AT** 11, 16; **INK 2**; **FLASH 1**;"."

Después de haber ejecutado la instruíción, el ordenador volverá a los valores normales o valores globales preestablecidos. PRINT también obedecerá los códigos locales de control del color, insertados conjuntamente con la información.

Formato

PRINT [TABexpr número entero;]
[AT expr número entero, expr número entero]
CHR$ (expr número entero);]
[sentencia;]lexpr numérica]
[expr de cadena][;][,][']

RANDOMIZE (RAND)

Posición en el teclado

T

Instrucción/Comando

La palabra clave RANDOMIZE se abrevia en el teclado a RAND y se utiliza conjuntamente con RND para generar una secuencia de números aleatorios y que por tanto no pueden predecirse. También puede ser una secuencia de números predecibles. (Es importante no confundir RAND y RND).

Cómo emplear RANDOMIZE

La palabra clave RANDOMIZE se utiliza dentro de una instrucción de un programa o bien como comando. Esta palabra clave se puede utilizar independientemente o bien seguida por un valor numérico, por ejemplo.

RANDOMIZE 1
10 **RANDOMIZE**

El valor numérico situado a continuación de RANDOMIZE se redondea (si es necesario) al número entero más próximo. Este valor puede variar entre 0 y 65535. Un valor superior a 0 ajusta el valor de la variable del sistema SEED a ese valor. El resultado es que la palabra clave RND siempre genera la misma sequencia de números. Dicha secuencia de números dependerá del valor de RANDOMIZE.

En cambio, cuando RANDOMIZE va seguida de un 0 o ningún valor, la variable de sistema SEED recibe el valor de otra variable de sistema denominada FRAMES. Esta variable FRAMES cuenta la cantidad de imágenes visualizadas por la televisión desde el mo-

mento en el que se conectó el INVES SPECTRUM+. SEED se incrementa cada 50 imágenes (véase la sección PAUSE). Debido a que SEED cambia cada 50 segundos, la secuencia de números generados por una instrucción que contenga RANDOMIZE y a continuación RND o solamente RANDOMIZE 0 ofrece un efecto sumamente aleatorio.

Formato

RANDOMIZE [expr número entero]

READ — Leer

Posición en el teclado

MODO EXTENDIDO
A

Instrucción/Comando

La palabra clave READ se utiliza en combinación con la palabra clave DATA y tiene como objetivo el asignar valores a variables, utilizando los valores especificados en la instrucción DATA.

Cómo emplear READ

Normalmente READ se utiliza para formar una instrucción dentro de un programa.

Debe ir seguida por una o más variables numéricas o variables de cadena *separadas entre sí por comas*. Por ejemplo.

20 **READ** a$,x

Cuando se ejecuta por primera vez la instrucción que contiene la palabra clave READ, el ordenador se dirigirá a la primera línea que contenga la palabra clave DATA y tomará la misma cantidad de valores que de variables, comenzando a partir del primer elemento en la lista de valores que aparecen a continuación de DATA. Los valores obtenidos en la lista de DATA se atribuyen a las variables de la instrucción READ sucesivamente y siguiendo el orden que tienen en la instrucción.

Formato

READ var-num [,var-num] [var de cadena]
READ var de cadena [,var-num] [,var de cadena]

REM REMark — Comentario

Posición en el teclado

E

Instrucción

REM tiene como objetivo el indicarle al ordenador que la información que aparece a continuación es solamente un comentario u óbservación. Esta puede consistir, por ejemplo, en el título y nombre del autor del programa, explicaciones acerca de las líneas siguientes en el programa, como por ejemplo el objetivo de una determinada variable. La instrucción REM se incluye en el programa como información al usuario, y no interviene en la operación del programa. Esta instrucción no se visualiza en la pantalla excepto cuando se hace un listado del programa (LIST).

Cómo emplear REM

REM forma parte de una línea independiente del programa o también puede incluirse como elemento final de otra línea. La palabra clave REM va seguida por los comentarios y no requiere comillas, por ejemplo:

10 **REM** El autor de este programa es Pedro
80 I**NPUT** n$; **REM** n$ es el nombre

Cuando el ordenador llegue a una línea que comience por la palabra clave REM ignorará el resto de la información que aparezca en di-cha línea.

Formato

REM [cualquier carácter]

RESTORE — Restaurar

Posición en el teclado

MODO EXTENDIDO
S

Instrucción/Comando

La palabra clave RESTORE utilizada junto con READ y DATA modifica el mecanismo normal de READ DATA. Su función es indicarle al ordenador que al ejecutar la instrucción READ, tome los valores de una determinada instrucción DATA, en lugar de seguir el procedimiento normal de dirigirse a la primera instrucción DATA del programa (véase DATA y READ).

Cómo emplear RESTORE

Por lo general RESTORE se incluye dentro de una instrucción que forma parte de un programa. El programador tiene la opción de agregar a continuación de ella algún valor unmérico, por ejemplo:

160 **RESTORE** 800

El valor numérico colocado a continuación de RESTORE se redondea, si es necesario al número entero más próximo. El ordenador supone que el valor numérico es el número de la línea de programa que contiene una instrucción DATA. Después de haber encontrado la palabra clave RESTORE, el ordenador procede de la manera siguiente: al encontrar una línea de programa con la palabra clave READ, se encaminará a la línea 800 del programa y tomará los valores de DATA especificados en ella para asignarlos a las variables de aquella línea. El procedimiento mediante el cual se

adjudica un valor a una variable es similar al utilizado con READ DATA (véase las secciones DATA y READ).

Si la línea de DATA (línea 800 en el ejemplo) no existe o no contiene una palabra clave DATA, entonces el ordenador continuará hasta la próxima sentencia DATA.

Cuando RESTORE vaya seguido por un número, o si no lo es por ningún valor numérico también, ocurre lo siguiente: el ordenador al encontrar una instrucción con la palabra clave READ avanza hasta la próxima línea que contenga la palabra clave DATA.

Formato

RESTORE [expr número entero]

RETURN — Regresar

Posición en el teclado
Y

Instrucción/Comando

La palabra clave RETURN se utiliza para terminar una subrutina e indicarle al ordenador que regrese al programa principal o a una subrutina previa.

Cómo emplear RETURN

Normalmente,RETURN se utiliza como parte de una instrucción dentro de un programa. RETURN se utiliza sólo (sin parámetros) si se coloca al final de la subrutina. Por ejemplo:

1080 **RETURN**

Al ejecutar esta instrucción el ordenador regresa a la sentencia inmediatamente después de la última palabra clave GOSUB ejecutada.

Para más detalles consúltese la sección sobre GOSUB.

Formato

RETURN

RND RanDom Number — Número aleatorio

Posición en el teclado

MODO EXTENDIDO
T

Función

Esta palabra clave se utiliza para generar un número aleatorio.

Cómo emplear RND

La palabra clave se utiliza sola (sin parámetros) formando parte de una instrucción de programa o como comando. Por ejemplo:

60 **LET** x=**RND**

Al ejecutar esta instrucción el ordenador genera un número al azar o aleatorio, menor que 1 y mayor o igual a 0.

Cuando se conecta el INVES SPECTRUM+ o bien se reinicializa, o bien se introduce la instrucción NEW, los números invocados por RND se producen en la misma secuencia. Esta secuencia se genera tomando las potencias de 75 (75,75*75, 75*75*75, y así sucesivamente) y después de dividir cada potencia por 65537 y tomar el resto, se sustrae de este 1 y se divide este resultado por 65536, etc.

Cuando se requiere una secuencia más aleatoria u otra secuencia predeterminada se incluirá la palabra clave RANDOMIZE antes de RND.

RANDOM y números enteros

Si se desea obtener números enteros entre 1 y x, se utilizará la palabra clave INT, en la fórmula INT(RND*X)+1. Para generar un número entero entre 0 y x, se utilizará la fórmula INT(RND*x+0.5).

Formato

RND

RUN — Ejecutar

Posición en el teclado
R

Comando/Instrucción

La palabra clave RUN le indica al ordenador que ejecute el programa, normalmente a partir de su primera línea.

Cómo emplear RUN

RUN puede emplearse como comando directo o puede formar parte de una instrucción dentro de un programa. Se puede incluir un valor numérico a continuación de RUN. Por ejemplo:

RUN 50

Cuando RUN no va seguido por un valor numérico, el ordenador ejecutará el programa a partir de la primera línea del programa. Cuando se incluye un valor numérico después de RUN, éste se redondeará, si es necesario al número entero más próximo, y el programa se ejecuta a partir de esa línea. Si el ordenador no encuentra una línea con ese número, el ordenador ejecutará el programa comenzando a partir de la línea inmediata siguiente.

Cuando se carga en la memoria del ordenador un programa almacenado en cinta mediante la instrucción LINE, el ordenador lo ejecutará inmediatamente, y no será necesario utilizar la instrucción RUN.

Formato

RUN [expr número entero]

SAVE — Grabar

Posición en el teclado

S

Comando/Instrucción

La palabra clave SAVE le indica al ordenador que registre en cinta el programa que tiene en su memoria con el fin de almacenarlo.

Cómo emplear SAVE

SAVE es utilizado por lo general como un comando directo, aunque también puede formar parte de una sentencia contenida en un programa.

SAVE «ProgramaX»

El nombre del programa puede contener hasta diez caracteres. Al ejecutarse esta instrucción, el ordenador visualizará en la pantalla el siguiente mensaje:

PREPARE LA CINTA Y PULSE ENTER

Una vez puesto en marcha el cassette, el usuario debe pulsar cualquier tecla y el ordenador comenzará a transmitir el contenido de su memoria RAM a la cinta. Al terminar de grabar, el ordenador visualiza en la pantalla el siguiente mensaje: 0OK, 0,1

Ejecución automática

Para indicarle al ordenador que ejecute automáticamente un programa pregrabado inmediatamente después de haberlo cargado en su memoria, se utilizarán las palabras clave SAVE y LINE en el momento de registrar en cinta el programa. La sintaxis es la siguiente: el nombre del programa, seguido por la palabra clave LINE y un valor numérico, por ejemplo:

SAVE «programaX» **LINE** 1

El valor numérico a continuación de LINE se redondeará (si es necesario) al número entero más próximo, y tendrá un valor 1 o el número de una línea de programa.

Al ejecutar estas instrucciones, el ordenador registrará el programa en cinta, como cuando ejecuta la instrucción SAVE.

Cuando el ordenador carga el programa pregabado, lo pondrá en ejecución automáticamente a partir de la línea indicada por el valor numérico. Si no encuentra una línea de ese número, el ordenador comenzará a ejectar el programa a partir de la línea siguiente en el programa.

Formato

SAVE expr de cadena [**LINE** expr número entero]

SAVE CODE

Posición en el teclado

S
MODO EXTENDIDO
I

Comando/Instrucción

La palabra clave SAVE CODE le indica al ordenador que envíe una determinada parte de la información almacenada en su memoria a la grabadora de cassettes, para que sea registrada en cinta y almacenada. Esta información podrá utilizarse nuevamente en el futuro. El procedimiento para cargar la información almacenada en cinta a la memoria del ordenador es utilizar la palabra clave LOAD CODE.

Cómo emplear SAVE CODE

La palabra clave SAVE CODE puede utilizarse como comando directo o bien incluirse en una instrucción, dentro de un programa. La primera parte de la palabra clave SAVE va seguida por el nombre que designará la información registrada en la cinta. Este nombre es un valor de cadena (string value). A continuación se inserta la segunda parte de la palabra clave: CODE. CODE debe ir seguido por dos valores numéricos separados por una coma Por ejemplo:

SAVE «dibujo»**CODE** 16384,6912

El nombre que identifica la información registrada en cinta, insertado entre SAVE y CODE puede tener un máximo de diez caracteres.

Los dos valores numéricos que siguen a CODE se redondearán (si es necesario) al número entero más próximo.

El primer número indica la dirección inicial (16384) de la información almacenada en la memoria. El segundo valor le indica al ordenador la cantidad de bytes que deben ser registrados (6912 bytes en el ejemplo).

El ordenador ejecuta esta instrucción y envía la información a la grabadora de cassettes para ser registrada, como cuando se ejecuta la instrucción SAVE. (Ver la sección sobre SAVE).

Formato

SAVE expr de cadena **CODE** expr de número entero, expr de número entero.

SAVE DATA — Grabar datos

Posición en el teclado

S
MODO EXTENDIDO
D

Instrucción/Comando

La palabra clave SAVE DATA le indica al ordenador que registre una matriz (array) en cinta magnética. Más tarde la matriz puede ser vuelta a cargar en el ordenador utilizando la palabra clave LOAD DATA.

Cómo emplear SAVE DATA

La palabra clave SAVE DATA puede utilizarse formando parte de una instrucción dentro de un programa o como comando directo. SAVE va seguida por el nombre bajo el cual se registró la matriz, a continuación se incluye la palabra DATA y luego una letra o una letra con el signo $, y finalmente un par de paréntesis vacíos.

450 **SAVE** «números» **DATA** n()
750 **SAVE** «nombres» **DATA**n$()

El nombre de la matriz puede incluir hasta un máximo de diez caracteres. La letra sola o con el signo $ colocada después de la palabra clave DATA será el nombre que identifica la matriz en el programa registrado en la cinta. La matriz se guarda en la cinta siguiendo el mismo procedimiento que en el caso de los programas registrados con la palabra clave SAVE.

Formato

SAVE expr de cadena **DATA** letra [$]()

SAVE SCREEN$ — Grabar pantalla

Posición en el teclado

S
MODO EXTENDIDO
CAMBIO K

Comando/Instrucción

SAVE SCREEN$ le indica al ordenador que registre la información visualizada en la pantalla en cinta magnética. En el futuro la información podrá ser cargada nuevamente en el ordenador mediante la instrucción LOAD SCREEN$.

Cómo emplear SAVE SCREEN$

La palabra clave SAVE SCREEN$ puede incluirse en una instrucción dentro de un programa o bien puede utilizarse como comando directo. El primer término, SAVE, va seguido por el nombre bajo el cual se registrará la imagen en la cinta. A continuación se incluye SCREEN$.

SAVE «imagen» **SCREEN$**

El nombre que identificará la imagen o infomación visualizada en la pantalla, registrada en el cassette, puede tener hasta diez caracteres.

Cuando el ordenador recibe esta instruccion, enviará la imagen o información a la grabadora para ser registrada en cinta, como cuando se ejecuta la palabra clave SAVE. (Ver la sección sobre SAVE).

Formato

SAVE expr de cadena **SCREEN$**
[**LINE** expr de número entero]

SCREEN$ — Pantalla

Posición en el teclado

MODO EXTENDIDO
CAMBIO K

Función

El propósito de la función SCREEN$ es detectar el carácter que aparece en una determinada posición de la pantalla.

Cómo emplear SCREEN$

La palabra clave SCREEN$ va seguida por dos valores numéricos separados por una coma y encerrados entre paréntesis, por ejemplo:

160 **IF SCREEN$** (l,c)=«x»**THEN PRINT** «CRASH!»

Los valores numéricos que siguen a SCREEN$ (redondeados si es necesario al número entero más próximo), son las coordenadas del punto en la pantalla. El primero de los valores (indicado en el ejemplo por la variable l), define el número de la línea horizontal sobre la cual se encuentra el punto, y puede variar entre 0 y 21. El segundo valor numérico (c) define la columna sobre la cual se encuentra el punto. Este valor puede oscilar entre 0 y 31.

Cuando el ordenador recibe esta instrucción, examinará el contenido de la posición especificada en la pantalla, y visualizará el carácter de esa posición como una constante de cadena (el caracter entre paréntesis, «x» en el ejemplo).

Si el ordenador no encuentra un carácter en la posición indicada entonces se limitará a visualizar una cadena vacía"".

Formato

SCREEN$ (expr número entero, expr número entero).

SGN SiGN — Signo

Posición en el teclado

MODO EXTENDIDO
F

Función

La palabra clave SIGN sirve para determinar si un número es positivo, negativo o cero.

Cómo emplear SGN

SGN debe ir seguido de un valor numérico, por ejemplo:

```
50 LET x=SGN y
```

Las expresiones deben ir *encerradas entre paréntesis*. Obedeciendo la palabra clave SGN, el ordenador visualizará un 1 si el argumento (en el ejemplo, y) es positivo, -1 si el argumento es negativo, y 0 si el argumento es 0.

Formato

SGN const-num
SGN var-num
SGN (expr-num)

SIN SINe — Seno

Posición en el teclado

MODO EXTENDIDO
O

Función

La función SIN suministra el valor del seno de un ángulo.

Cómo emplear SIN

SIN debe ir seguido por un valor numérico, por ejemplo:

```
80 LET x=SIN y
```

Las expresiones deben ir encerradas entre paréntesis. El valor numérico que se coloca a continuación de SIN es el ángulo en radianes, y la función SIN indica cuál es el seno correspondiente a ese ángulo.

Los grados pueden convertirse en radianes multiplicándolos por P1/1 80.

Observe que SIN suministra un valor positivo para ángulos de entre 0 y 180 grados, y un valor negativo para ángulos de entre 180 y 360 grados.

Formato

SIN const-num
SIN var-num
SIN (expr-num)

SQR SQuare Root — Raíz cuadrada

Posición en el teclado

MODO EXTENDIDO
H

Función

SQR calcula la raíz cuadrada de un número.

Cómo emplear SQR

La palabra clave SQR debe ir seguida por un valor numérico, por ejemplo

```
70 LET x=SQRy
```

Las expresiones deben ir *encerradas entre paréntesis*. El valor que aparece a continuación de SQR (en el ejemplo, y) debe ser siempre mayor que cero, y la función producirá su raíz cuadrada.

Formato

SQR const-num
SQR var-num
SQR (expr-num)

STOP

Posición en el teclado

CAMBIO A

Instrucción/Comando

STOP detiene la ejecución del programa en un punto determinado. Por ejemplo, esta palabra clave puede utilizarse para terminar la parte principal de un programa, lo que permite incluir subrutinas separadas, a continuación de la parte principal. STOP también es un auxiliar útil cuando se aplican técnicas para detectar y corregir los errores en los programas.

Cómo emplear STOP

Normalmente STOP forma parte de una instrucción, dentro de un programa. Se utiliza solo (sin ningún parámetro), por ejemplo:

```
350 STOP
```

Durante la ejecución del programa el ordenador se detendrá al llegar a esta línea y producirá el siguiente mensaje de error en la pantalla:

9 SENTENCIA STOP

agregando el número de la línea y de la sentencia donde se ha detenido el programa.

Si se introduce la instrucción CONTINUE el ordenador reanudará el programa a partir de la sentencia siguiente, con los nuevos valores.

Formato

STOP

**STR$** STRing$ — Cadena

Posición en el teclado

MODO EXTENDIDO
Y

Función

La palabra clave STR$ convierte números en cadenas (string).

Cómo emplear STR$

STR$ va seguido de un valor numérico, por ejemplo:

```
90 LET a$=STR$ x
```

Las expresiones deben estar encerradas entre paréntesis. Al ejecutar la instrucción STR$, el ordenador visualizará en la pantalla el valor de su argumento (el número) en forma de constante de cadena.

En el ejemplo, si se asigna un valor de 65 a x, la instrucción asignará un valor de «65» a a$.

Formato

STR$ const-num
STR$ var/num
STR$ (expr-num)

TAB Ver LPRINT, PRINT

TAN Tangente

Posición en el teclado

MODO EXTENDIDO
E

Función

TAN suministra la tangente de un ángulo.

Cómo emplear TAN

La palabra clave TAN va seguida por un valor numérico, por ejemplo:

130 **LET** x=**TAN** y

Las expresiones deben estar entre paréntesis. El valor a continuación de TAN es el ángulo en radianes. TAN obtiene la tangente del ángulo.

Los grados pueden convertirse en radianas multiplicándolos por P1/1 80.

Formato

TAN const-num
TAN var-num
TAN (expr-num

TO Hasta

Posición en el teclado

CAMBIO
F

Función

La palabra clave TO tiene dos usos diferentes en el dialecto de BASIC utilizado por INVES SPECTRUM+.

En primer lugar, TO puede ser utilizado conjuntamente con la palabra clave FOR para construir un bucle FOR NEXT (ver la sección FOR por más detalles).

En segundo lugar, TO también puede utilizarse para fragmentar (slicing) cadenas (strings). La fragmentación es la operación de dividir cadenas en subcadenas más pequeñas.

Cómo se emplea TO para fragmentar cadenas

En este caso, TO se utiliza para designar el primer carácter y el último carácter de la subcadena contenida en la cadena principal. La palabra clave TO va precedida por una variable de cadena, a continuación se abre un paréntesis seguido de un valor numérico. Después de TO se agrega otro valor numérico y se cierra el paréntesis. Por ejemplo:

80 **PRINT** a$ (4 TO 7)

Las expresiones de cadena (string expresión) deben también *encerrarse entre paréntesis*. El valor de cadena (en el ejemplo a$) identifica la cadena que se desea fragmentar. Los dos valores numéricos (en el ejemplo los números 4 y 7) le indican al ordenador las posiciones del primer y último carácter de la subcadena dentro de la cadena principal. Al ejecutar la instrucción TO, el ordenador tomará el fragmento definido en la instrucción de la cadena principal y visualizará la subcadena (caracteres 4 a 7 de a$).

Si no se incluye un número identificando el comienzo de la subcadena, el ordenador asumirá que el valor numérico es 1. De la misma manera, si no se incluye un segundo número, para indicar dónde termina la subcadena, entonces el ordenador asume que ésta termina en el último carácter de la cadena principal.

Formato

const de cadena ([expr-núm) **TO** [expr-num]
var de cadena ([expr-núm] **TO** [expr-num])
(expr de cadena) ([expr-núm] **TO** [expr-num])

USR User SubRoutine — Subrutina del usuario

Posición en el teclado

MODO EXTENDIDO
L

Función

La palabra clave USR se emplea para llamar a una subrutina en código máquina. Esta subrutina está almacenada en una dirección específica de la memoria del ordenador. USR también puede utilizarse para depositar los datos (data) de los gráficos definidos, por el usuario, en las posiciones reservadas al final de la memoria alta del ordenador.

USR y el código máquina

Para utilizar el código máquina, USA debe ir seguido por un valor numérico, por ejemplo:

80 **PRINT USR** 65000
100 **RANDOMIZE USR** 65000

Las expresiones deben ir entre paréntesis. El valor numérico que aparece a continuación de la palabra clave USR se redondeará al número entero más próximo. Este valor es la di-

rección inicial en la memoria, donde está almacenada la subrutina en código máquina. Cualquier instrucción del programa que contenga USR llamará a dicha subrutina, y USR producirá el valor del registro bc.

USR y los gráficos definidos por el usuario

Para crear gráficos definidos por el usuario se empleará la palabra clave USR junto con la palabra clave POKE. USR debe ir seguida por una constante de cadena o una variable para invocar una dirección para la instrucción que contenga POKE. Por ejemplo:

5 **POKE USR** «a»,255

El valor de cadena insertado a continuación de USR puede ser una sola letra, en minúsculas o mayúsculas entre A y U. El ordenador no distinguirá entre mayúsculas y minúsculas.

USR también suministrará la dirección inicial (dirección del primer byte) de una de las 21 zonas de la memoria especialmente reservadas para los gráficos definidos por el usuario. Cada zona contiene ocho bytes y para crear un carácter gráfico utilizando la palabra clave POKE, se puede almacenar un número directamente en una posición de la memoria.

Formato

USR const número entero
USR var número entero
USR (expr número entero)
USR const de cadena
USR var de cadena

VAL VALue — Evaluar

Posición en el teclado

MODO EXTENDIDO
J

Función

La palabra clave transforma una cadena con un valor numérico en un número.

Cómo emplear VAL

VAL debe ir seguida .por una constante o variable de cadena. Por ejemplo:

70 **LET** x=**VAL** a$

El ordenador elimina las comillas del valor de la constante o variable de cadena, y una vez hecho esto evalúa la expresión como valor numérico. Cuando los elementos han sido evaluados, VAL devuelve un resultado numérico.

Ejemplos

Si a$ tiene un valor de «435», la instrucción del ejemplo anterior asigna un valor de 435 a la variable x. VAL puede además evaluar expresiones, por ejemplo:

10 **INPUT** a$,x
20 **PRINT VAL** a$

El valor de cadena que se asigne a a$ debe ser una expresión que lleve x, por ejemplo x*x. El ordenador asigna un valor numérico a x, por ejemplo 5. VAL elimina las comillas para obtener x*x y obtiene un resultado utilizando el valor asignado a x. Por ejemplo, si se asignó a x un valor 5, el resultado será 5*5=25.

Formato

VAL const de cadena
VAL var de cadena

VAL$ VALue (string) — Evaluar (cadena)

Posición en el teclado

MODO EXTENDIDO
CAMBIO J

Función

VAL$ evalúa una cadena como una expresión de cadena. Su resultado es una cadena, como en caso de VAL, pero en lugar de producir un valor numérico como este último. VAL$ produce otra cadena.

Cómo emplear VAL$

VAL$ debe ir seguido por una variable de cadena, por ejemplo:

130 **PRINT VAL$** a$

El ordenador elimina las comillas del valor de la variable de cadena y evalúa la expresión resultante como una cadena. VAL$ calcula el valor de la expresión y produce un resultado en forma de constante de cadena.

Ejemplos

Pruebe este programa

10 **INPUT** a$,x$
20 **PRINT** VALS a$

El valor de cadena que se asigna a a$ debe ser una expresión que emplee x$, por ejemplo «x$+x$». El ordenador evalúa x$ como una cadena, por ejemplo «DO». La palabra clave VAL elimina las comillas del valor atribuido a a$ para obtener x$+x$, y lo evalúa utilizando el valor que se asignó a x$. En nuestro caso, el ordenador visualizará como resultado DODO.

Formato

VAL$ var de cadena

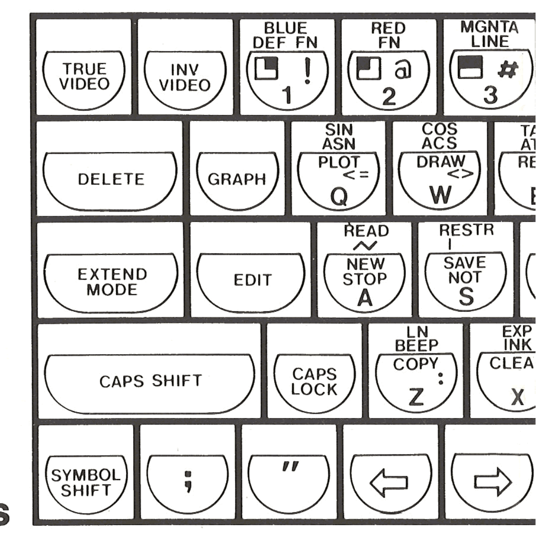
VERIFY — Verificar

Posición en el teclado

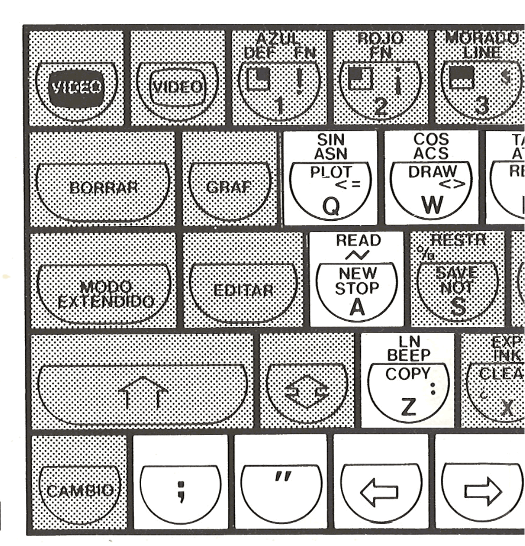
MODO EXTENDIDO
CAMBIO R

APENDICE A

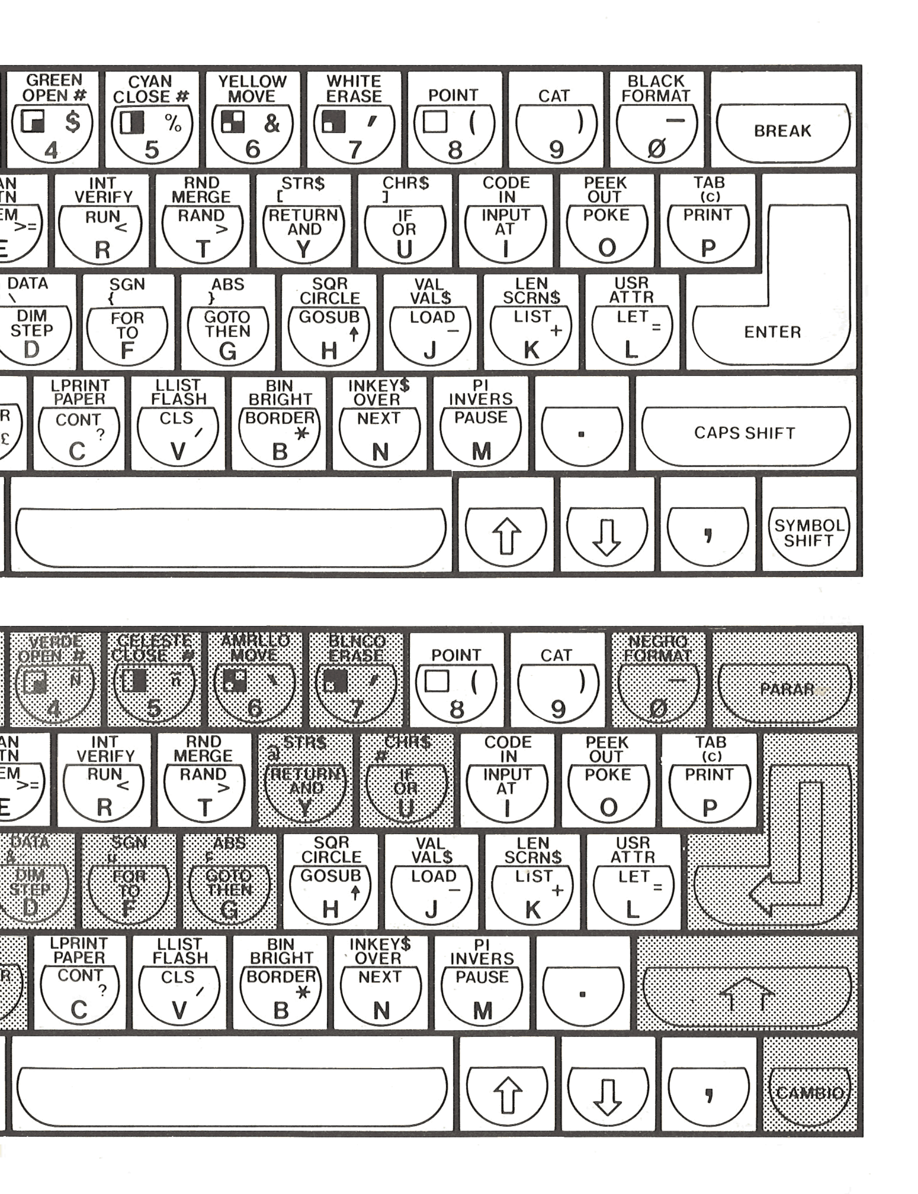
Relación entre los teclados inglés y español



Teclado inglés



Teclado español



GREEN OPEN # $ 4
CYAN CLOSE # % 5
YELLOW MOVE & 6
WHITE ERASE ' 7
POINT (8
CAT) 9
BLACK FORMAT — Ø
BREAK
INT VERIFY RUN < R
RND MERGE RAND > T
STR$ [RETURN AND Y
CHR$] IF OR U
CODE IN INPUT AT I
PEEK OUT POKE O
TAB (c) PRINT P
DATA \ DIM STEP D
SGN { FOR TO F
ABS } GOTO THEN G
SQR CIRCLE GOSUB ↑ H
VAL VAL$ LOAD — J
LEN SCRN$ LIST + K
USR ATTR LET = L
ENTER
LPRINT PAPER CONT ? C
LLIST FLASH CLS / V
BIN BRIGHT BORDER * B
INKEY$ OVER NEXT N
PI INVERS PAUSE M
.
CAPS SHIFT
,
SYMBOL SHIFT
VERDE OPEN # 4
CELESTE CLOSE # 5
AMRLLO MOVE 6
BLNCO ERASE 7
POINT (8
CAT) 9
NEGRO FORMAT Ø
PARAR
INT VERIFY RUN < R
RND MERGE RAND > T
STR$ RETURN AND Y
CHR$ IF OR U
CODE IN INPUT AT I
PEEK OUT POKE O
TAB (c) PRINT P
DATA DIM STEP D
SGN FOR TO F
ABS GOTO THEN G
SQR CIRCLE GOSUB ↑ H
VAL VAL$ LOAD — J
LEN SCRN$ LIST + K
USR ATTR LET = L
LPRINT PAPER CONT ? C
LLIST FLASH CLS / V
BIN BRIGHT BORDER * B
INKEY$ OVER NEXT N
PI INVERS PAUSE M
.
,
CAMBIO

APENDICE B

Mensajes de Error

Cuando el ordenador INVES SPECTRUM+ deja de escribir BASIC, aparece un mensaje al fondo de la pantalla de televisión. Este mensaje le indica que se ha completado un programa, o un comando, o bien que se ha cometido un error. Estos mensajes consisten en un número de código o una letra seguidos por un mensaje breve, y a continuación los números de la línea y la instrucción donde se paró el ordenador. Los comandos se indican como línea 0, y dentro de una línea la instrucción 1 se localiza al comienzo de ésta, la instrucción 2 se coloca después de los primeros dos puntos o THEN y así sucesivamente. CONTINUE por lo general hace que el programa- comience de nuevo en la línea que se especificó en el mensaje.

0 OK
Se ha completado con éxito una operación determinada. Si se incluye la palabra clave CONTINUE, el ordenador ignorará este último informe y reanudará la ejecución del programa a partir de la instrucción especificada en el mensaje anterior.

1 **NEXT sin FOR**
Al ejecutar el programa, el ordenador ha encontrado una palabra clave NEXT pero no ha podido encontrar su palabra clave complementaria FOR aunque existe una variable ordinaria con el mismo nombre.

2 Variable indefinida
Se ha utilizado una variable de control con la palabra clave NEXT, pero sin incluir previamente su palabra clave complementaria FOR.

Se ha incluido una variable con subíndice pero sin haber definido sus dimensiones mediante una instrucción DIM, o sin haber cargado la matriz desde la cinta magnética.

3 Indice erróneo
El subíndice de una variable se encuentra fuera de las dimensiones de la matriz correspondiente.

4 Sin memoria
No queda suficiente espacio en la memoria del ordenador para completar la sentencia o comando.

5 Fuera de pantalla
Una sentencia INPUT ha generado más de 23 líneas en la mitad inferior de la pantalla, o se ha utilizado un número de línea de 22 o más con la palabra clave PRINT AT.

6 Número muy alto
El ordenador ha intentado producir un número mayor que 10(38) aproximadamente.

7 RETURN sin GOSUB
El ordenador ha encontrado una palabra clave RETURN pero no puede localizar la palabra clave complementaria GOSUB correspondiente.

8 Fin de fichero
Mensaje que informa sobre la manipulación de información registrada en el Microdrive.

9 Sentencia STOP
El ordenador ha encontrado una instrucción que contiene la palabra clave STOP. Con la palabra clave CONTINUE se reanudará el programa a partir de la instrucción siguiente.

A Argumento erróneo
Se ha dado un argumento o valor incorrecto.

B Entero fuera de rango
El ordenador ha redondeado un número al número entero más próximo. El resultado queda fuera de los límites del ordenador y no puede ser aceptado por éste.

C No existe en BASIC
El ordenador ha encontrado una sentencia que no tiene sentido en BASIC y en el contexto del programa.

D BREAK-CONT sigue
Se ha oprimido la tecla PARAR. El efecto de la palabra clave CONTINUE es repetir la instrucción a partir del lugar en que se detuvo eh ordenador.

E Faltan datos
Obedeciendo una instrucción READ, el ordenador se ha dirigido a la lista de datos o información contenida en una sentencia encabezada por la palabra clave DATA.

F Nombre/fichero erróneo
SAVE ha encontrado que el nombre agregado a continuación de SAVE tiene más de diez caracteres.

G No hay sitio
No hay suficiente sitio en la memoria del ordenador para admitir una nueva línea de programa.

H STOP en INPUT
Al ejecutar una instrucción INPUT, el ordenador ha encontrado la palabra clave STOP. La palabra clave CONTINUE repite la instrucción INPUT.

I FOR sin NEXT
El ordenador ha encontrado un bucle FOR... NEXT cuyos límites o los valores definidos por STEP no son correctos. Por ejemplo: FOR x = 5 TO 4, pero no se ha encontrado la palabra NEXT complementaria.

J Dispositivo E/S erróneo
Mensaje sobre la manipulación de información registrada en un Microdrive.

K Color Inválido
El valor que se especificó en instrucciones que incluyen palabras clave INK, PAPER, FLASH, INVERSE y OVER o el carácter de control correspondiente se encuentra fuera de los límites del ordenador.

L BREAK en programa
Durante la ejecución de un programa se ha oprimido la tecla PARAR. Si se desea continuar ejecutando el programa bastará con introducir la palabra clave CONTINUE.

M RAMTOP mala
El valor especificado para RAMTOP es demasiado grande o demasiado pequeño.

N Sentencia perdida
Durante la ejecución del programa el ordenador ha encontrado una instrucción indicándole que avance a una sentencia que ya no existe.

O Corriente inválida
Información registrada en un Microdrive.

P FN sin DEF
La palabra clave FN estaba sin su correspondiente DEF FN.

Q Parámetro erróneo
La instrucción que contiene la palabra clave FN contiene un número equivocado de valores o argumentos que deben pasarse a la función. (Una cadena en lugar de números o viceversa.)

R Error de carga
La información o verificación no se ha completado correctamente. El ordenador no especifica cuál ha sido el error.

APENDICE C

Más allá del BASIC

El BASIC es un lenguaje de uso general para ordenadores que funciona adecuadamente en la mayoría de las aplicaciones. El usuario puede disponer además de otros programas (software) que le suministran otros lenguajes, como por ejemplo FORTH, microPROLOG y LOGO. Estos lenguajes operan de forma muy diferente al BASIC y le abren nuevas posibilidades a su ordenador.

Como el BASIC es un lenguaje muy general, a veces puede resultar incómodo o no muy adecuado a ciertas aplicaciones especiales. También es comparativamente lento. Otros lenguajes pueden ofrecer una mayor flexibilidad combinada con una programación más sencilla y velocidades mayores. El lenguaje FORTH, por ejemplo, le permite definir sus propias palabras en la programación. Estas palabras se incluirán en instrucciones que su ordenador comprenderá y ejecutará a velocidades diez veces superiores a las de los comandos equivalentes en el lenguaje BASIC.

En el caso del lenguaje micro-PROLOG, el ordenador adquiere la capacidad de comprender fráses inglesas sencillas, almacenandalas en su memoria como base para desarrollar un diálogo con el usuario. El LOGO es un lenguaje para ordenadores desarrollados con fines educacionales.

El código máquina

El lenguaje BASIC tiene como objetivo fundamental el permitir que el usuario pueda darle instrucciones al ordenador en una forma fácil de comprender.

Unidad Central de Proceso. Las instruíciones en BASIC que se introducen al ordenador, por ejemplo desde el teclado se traducen mediante un programa permanente llamado intérprete de BASIC (Basic Interpreter). Este programa se encuentra almacenado en una zona de la memoria del ordenador y convierte cada instrucción en BASIC es una secuencia de señales codificadas, que se envían a continuación a la CPU.

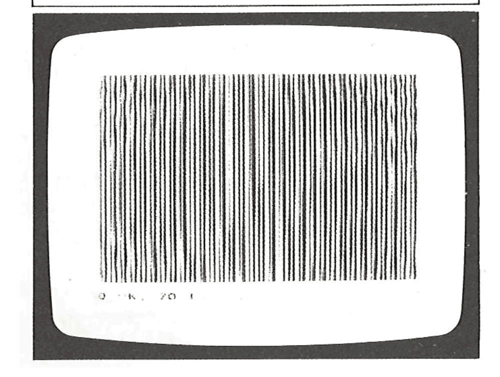
El intérprete toma algún tiempo en traducir las instrucciones en BASIC al lenguaje (código máquina) que comprende la CPU Z80.

También se puede evitar el uso del intérprete enviando las instrucciones redactadas en código máquina directamente a la Z80A en la CPU del ordenador. Si se hace así, el programa se ejecutará con gran rapidez.

Para tener una idea acerca de la rapidez con que el ordenador ejecuta instrucciones en código máquina, le sugerimos que introduzca en el ordenador, a través del teclado, el programa siguiente:

FRANJAS INSTANTANEAS

```
  10 FOR x=0 TO 15
  20 READ n: POKE 65000+x,n
  30 NEXT x
  40 DATA 33,255,63,1,1,24,22
  50 DATA 55
  60 DATA 35,11,120,177,200,114,
24,248
  70 RANDOMIZE USR 65000
```



Pruebe a cambiar el valor numérico 55, de la línea 50 del programa a cualquier valor entre 1 y 255. Con ello cambiará las franjas.

También se pueden producir franjas de colores insertando comandos que contengan la palabra clave INK. Lo que deseamos es demostrarle la velocidad con que cambian las imágenes producidas por el ordenador en la pantalla cuando las instrucciones se redactan en código máquina.

Este programa funciona porque las instrucciones DATA contienen 16 códigos que se almacenan a partir de la dirección 65000 de la memoria, en ejecución de las instrucciones 10 a la 30 del programa. La línea 70 envía los códigos al circuito integrado (microchip Z80A) produciendo inmediatamente la imagen en la pantalla.

Para ayudarle a redactar sus propios programas en código máquina, existen programas especiales denominados ensambladores (assemblers). Los ensambladores operan con una lista de instrucciones para ser introducidas en el ordenador, en lugar de la lista de números utilizada por el código máquina. El propósito del ensamblador es traducir estas instrucciones a los números del código máquina. Las instrucciones del ensamblador no son palabras clave en inglés como en BASIC.

APENDICE D Terminología de los ordenadores.

A continuación explicaremos algunos de los significados de los términos que aparecen en esta publicación, y que están asociados con el manejo de los, ordenadores.

Las palabras en bastardilla se han explicado en otra parte de esta sección. Si desea averiguar el significado de algún término que no se encuentra a continuación, le sugerimos que consulte el índice.

Almacenar (save). Procedimiento mediante el cual se guarda un programa o datos en un sistema de almacenamiento de datos, como puede ser un cassette o un cartucho de Microdrive.

Anidar o insertar (nesting). Ese término se aplica en el caso de programas con bucles, cuando uno o más bucles se encuentran «anidados» en el interior de otro bucle.

Argumento (argument). Es un valor, empleado por una función para obtener un resultado.

Atributos (attributes). Códigos que le atribuyen colores a los caracteres.

BASIC. Es el idioma de ordenadores utilizado por el INVES SPECTRUM+ y la mayoría de los microordenadores utilizados en el ho-gar

Binario, **Código** (Binary Code). Es un tipo de código utilizado por los ordenadores.

Bit. Uno de los dos elementos fundamentales del código binario.

Bucle (loop). Es una parte de un programa que se repite un cierto número de veces.

Byte. Es un conjunto de ocho bits. Representa un número. El valor de este número puede variar entre 0 y 255. Cada una de las direcciones en la memoria del ordenador alberga un byte.

Cadena (string). Es un grupo de uno o más caracteres, encerrados entre comillas para distinguirlos de los números y variables numéricas.

Cadena, Variable de (string variable). Variable que contiene una cadena. Las variables de cadena consisten siempre en una sola letra seguida por un signo de dólar, $.

Carácter. Cualquier letra, número (0 a 9) o unidad gráfica que el ordenador pueda imprimir o visualizar en la pantalla.

Caracteres, conjunto de (character set). Conjunto de los caracteres y ciertos códigos

de control preestablecidos que emplea el ordenador.

Cargar (load). Introducir en el ordenador un programa o datos a partir de un equipo de almacenamiento, que puede ser, por ejemplo una grabadora de cassettes o un *cartucho rom*.

Código máquina es el lenguaje de programación que su INVES SPECTRUM+ entiende. Los programas redactados en BASIC son traducidos al código máquina por el ordenador, antes de ejecutarlos.

Comando (command). Una instrucción individual que es ejecutada por el ordenador, o un comando directo.

Comando directo (direct command). Conjunto de una o más instrucciones que el ordenador ejecuta inmediatamente.

Constante (constant). Un número o un grupo formado por una o más letras o cualquier otro tipo de caracteres.

CPU (Central Processing Unit). Ver Unidad Central de Proceso.

Cursor. Es una letra parpadeante que indica la posición en la pantalla donde se visualizará algún elemento. Puede indicar el modo en el cual se encuentra el ordenador en ese momento.

Datos (Data). Son los datos transferidos al ordenador por un programa, o por otros métodos, para producir resultados.

Dirección (address). Unidad individual de memoria. Existen 65536 direcciones en la memoria del INVES SPECTRUM+.

Editar (edit). Es la operación de modificar algunos detalles de un programa.

Expresión (expression). Una combinación de constante, variables, y palabras claves.

Función (function). Es una operación donde el ordenador toma uno o más valores (o argumentos) y los utiliza para obtener un resultado. El resultado es otro valor.

Gráficos (graphics). Imágenes producidas por el ordenador y que pueden ser figuras, esquemas o diagramas.

Hardware (equipos). El ordenador y otras máquinas accesorias, por ejemplos los periféricos.

Imprimir (print). Visualizar resultados o gráficos, bien en la pantalla de televisión o bien impresos, utilizando para ello una impresora.

Información (information). Palabras, números y signos que el ordenador debe manipular, bien individualmente o combinados.

Input. Programas y datos alimentados o introducidos en el ordenador.

Interface. Unidad que conecta el ordenador con componentes periféricos (o los periféricos entre ellos mismos) para asegurar la comunicación de unos con los otros.

Introducir (enter). Es la operación por la cual se comunica al ordenador una instrucción o información completa.

K. Es una medida de capacidad de la *memoria* de un ordenador. 1K es equivalente a 1 kilobyte o 1024 bytes. La capacidad de la memoria indicada en K, es la cantidad total de direcciones en la memoria. Cada una de estas direcciones puede albergar un byte. El INVES SPECTRUM+ tiene una memoria de 48K RAM y 16K ROM, lo que ofrece una capacidad total de 64K.

Línea (line). Instrucción o conjunto de instrucciones de un programa. Cada línea lleva un número que le indica al ordenador la secuencia que debe seguir cuando ejecuta el programa.

Listado (listing). Una lista ordenada de las líneas de programa.

Lógico (logic). Es el proceso mediante el cual un ordenador decide si los resultados son correctos o incorrectos, o determina si determinados elementos son *verdaderos* o *falsos*.

Matriz (array). Grupo de unidades de información, vinculadas entre ellas y que se guardan como un grupo en la memoria del ordenador.

Memoria (memory). Es la parte del ordenador que almacena el *programa*, los datos y también las instrucciones de operación permanentes.

Mensaje. El ordenador lo visualiza en la pantalla con el fin de informar sobre las acciones que está llevando a cabo.

Modo (mode). En el INVES SPECTRUM+ uno de los cinco estados que determinan cuáles serán las palabras clave y los caracteres producidos por cada tecla en el teclado del ordenador.

Operador (operator). Una instrucción que ejecuta operaciones aritméticas o *lógicas*.

Output (salida o rendimiento). Son los resultados producidos por el ordenador.

Palabra clave (keyword). Instrucción en BASIC para el ordenador.

Periférico (peripheral). Cualquier accesorio conectado a su ordenador.

Pixel. Es el punto de color más pequeño que puede aparecer en la pantalla de su televisión.

Programa (program). Una secuencia de instrucciones para ser ejecutadas por el ordenador.

RAM (Random Access Memory). Es una zona de la *memoria* total del ordenador que puede almacenar programas o datos, y otros valores variables. También se le conoce con el nombre de memoria volátil, ya que su contenido se borra al desconectar el ordenador. El ordenador INVES SPECTRUM+ tiene una memoria RAM con una capacidad de 48K.

Registro (register). Pequeña unidad de memoria, separada de la *memoria principal*. Los registros que se encuentran en la CPU (Unidad Central de Proceso) se utilizan para la ejecución de los procesos de cálculo.

Resolución (resolution). Nitidez o cantidad

de detalles que pueden producirse en los gráficos del ordenador.

ROM (Read Only Memory). Zona de la memoria total del ordenador y que contiene *programas* e instrucciones permanentes. Esta memoria sólo puede leerse. El ordenador INVES SPECTRUM+ tiene una memoria ROM de 16K.

Save (ver almacenar).

Scroll (enrollar). Movimiento vertical de los caracteres que aparecen en la pantalla de televisión, que permite la aparición de más información.

Sentencia (statement). Palabra clave utilizada para formar una instrucción de una línea de programa o la instrucción misma.

Sintaxis (syntax). Es la secuencia correcta de las *palabras clave*, *constantes*, *variables* y *expresiones*, requeridas para formar una instrucción válida en BASIC.,

String (ver: cadena).

Unidad Central de Proceso (Central Processign Unit o CPU). Es el elemento central de su ordenador. Ejecuta las funciones de cómputación y controla las demás unidades. El ordenador INVES SPECTRUM+ utiliza un microprocesador Z80A.

Valor (value). Cualquiera número o grupo de constantes, *variables* o *expresiones* requeridos del ordenador o producidos por éste.

Variable (variable). Una o más unidades de memoria que contienen una determinada constante para que el ordenador la utilice. El INVES SPECTRUM+ distingue entre *variables numéricas* y *variables de cadena*.

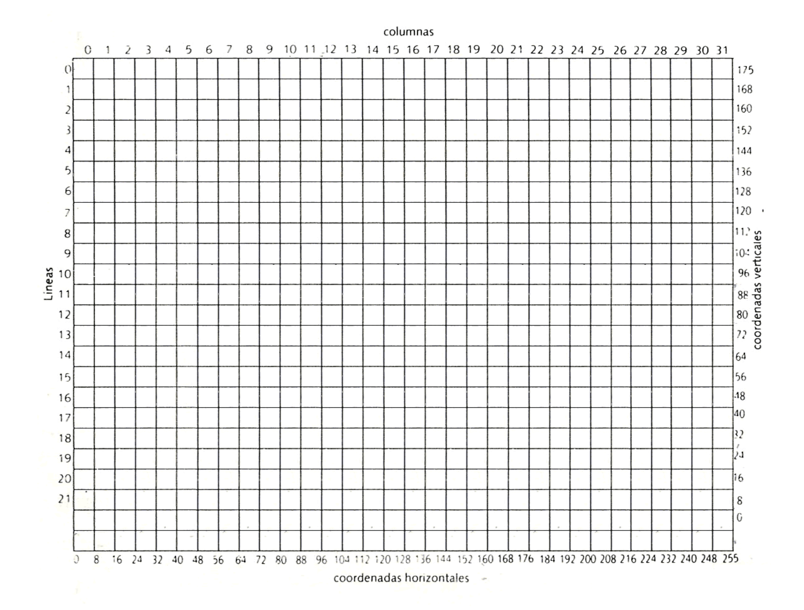
Cuadrícula para la presentación visual en la pantalla de televisión

La cuadrícula que damos a continuación muestra las coordenadas que se deben utili-zar en gráficos de alta y baja resolución. Aun-que las palabras clave individuales que con-trolan los efectos gráficos operan sólo con coordenadas de alta o baja resolución, cuando se produce la presentación visual se pueden utilizar ambas al mismo tiempo.

La cuadrícula de baja resolución se divide en dos partes principales, el área de visualización principal y dos líneas en la parte inferior de la pantalla. La palabra clave PRINT AT producirá caracteres en la parte superior y la parte clave INPUT AT producirá palabras en la parte inferior.

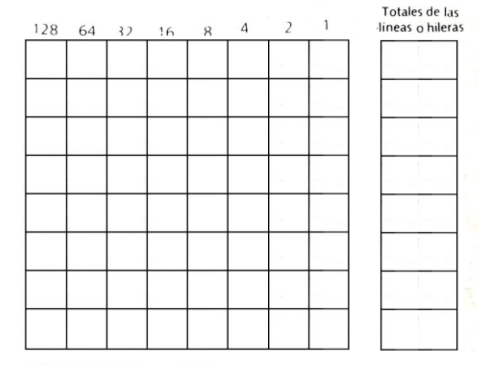
Las coordenadas para la pantalla de baja resolución se indican sobre los márgenes superior y de la izquierda de la cuadrícula.

La cuadrícula de alta resolución ocupa únicamente el área de visualización principal. Las palabras clave PLOT, DRAW y CIRCLE se emplean para producir gráficos de alta resolución. Las coordenadas de la cuadrícula de alta resolución se indican sobre los márgenes inferior y derecho de la cuadrícula.



Cuadrícula para los caracteres gráficos definidos por el usuario

Deben sumarse los números de la tabla para obtener el valor decimal de cada línea o fila horizontal de la cuadrícula. A continuación, para programar el carácter, se utilizará la palabra clave POKE USR.



investronica

Tomas Breton 62
Tel (91) 467 82 10
Telex 23399 IYCO E
28045 Madrid

Camp. 80
Tels. (93) 211 26 58 - 211 27 54
08002 Barcelona

EL MATERIAL EXHIBIDO
ES SOLO PARA USO
EDUCATIVO, NO COMERCIAL


ENCICLOPEDIA DE
REFERENCIA | EDUCATIVA
RETRO TECNIA
PARA ENTENDER LA EVOLUCION DE LOS ORDENADORES A TRAVES DEL TIEMPO